NEW 286s FROM AST, DELL, IBM ▶

FORTRAN READY FOR OS/2

DECEMBER 1988 VOL. 6 NO. 12 $3.95

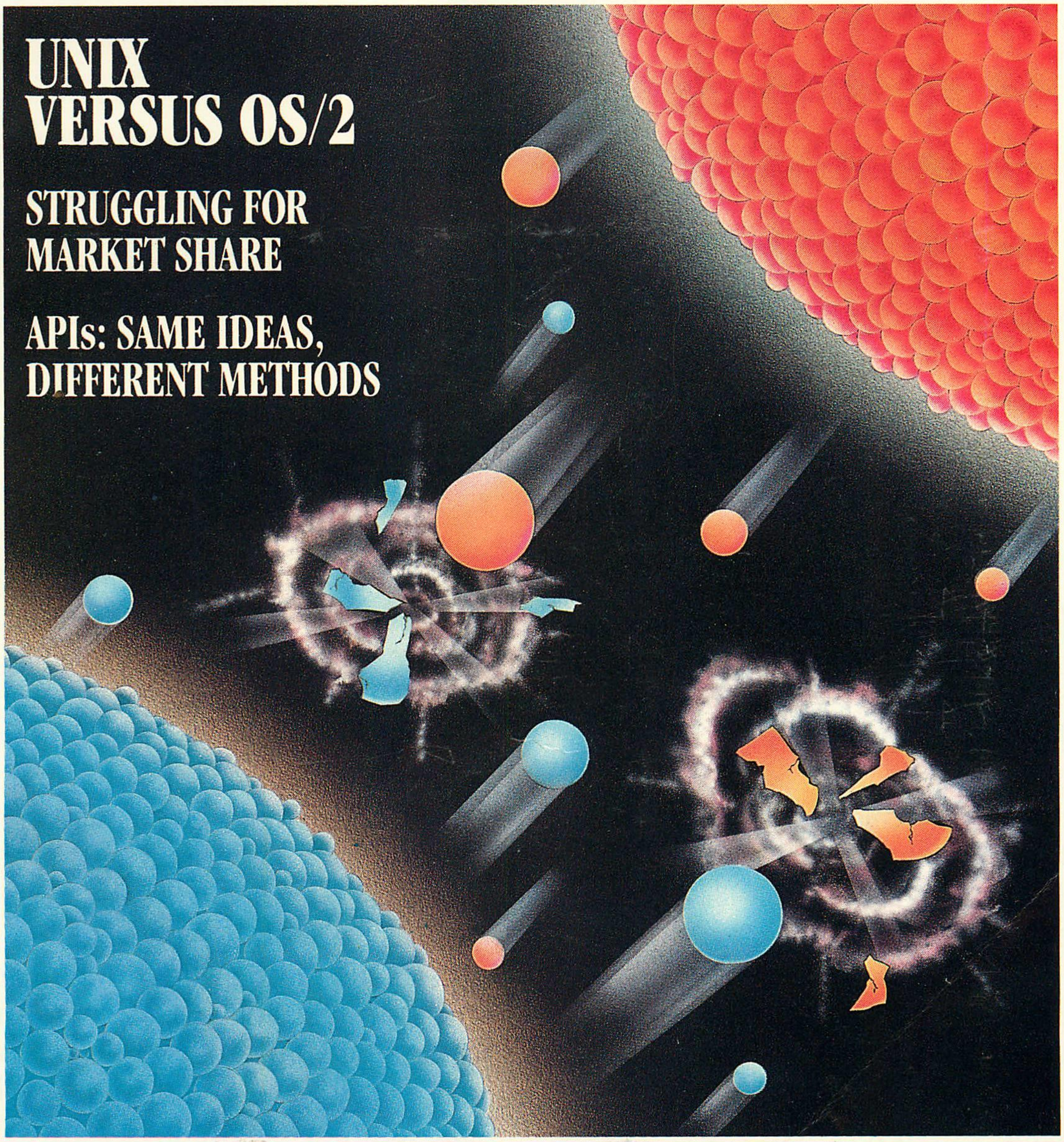
PC TECH JOURNAL®

FOR SYSTEMS DEVELOPERS AND INTEGRATORS

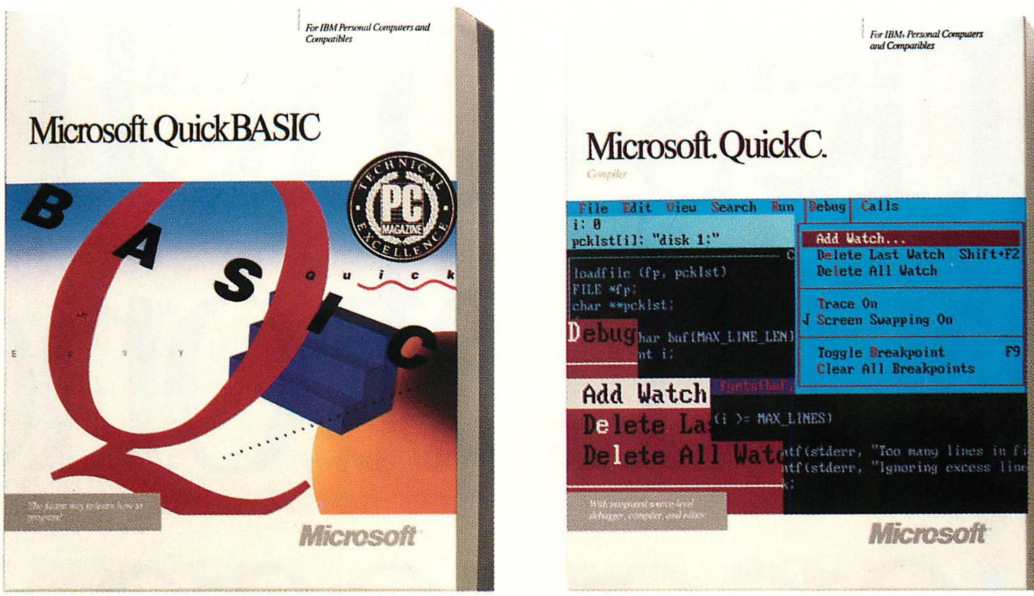
UNIX VERSUS OS/2

STRUGGLING FOR MARKET SHARE

APIs: SAME IDEAS, DIFFERENT METHODS



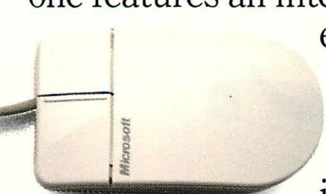
To each



Quick.

Despite what some software companies are saying, one language just can't be all things to all programmers. Which is why Microsoft offers a language for every level.

When you're starting out, you need learning tools that make it easier. Take our Quick languages, for instance. Each one features an integrated development environment that puts a powerful editor, compiler, debugger and on-line help within easy reach. And of course, all Quick languages are mouse-driven, so all commands are just a point and a click away.

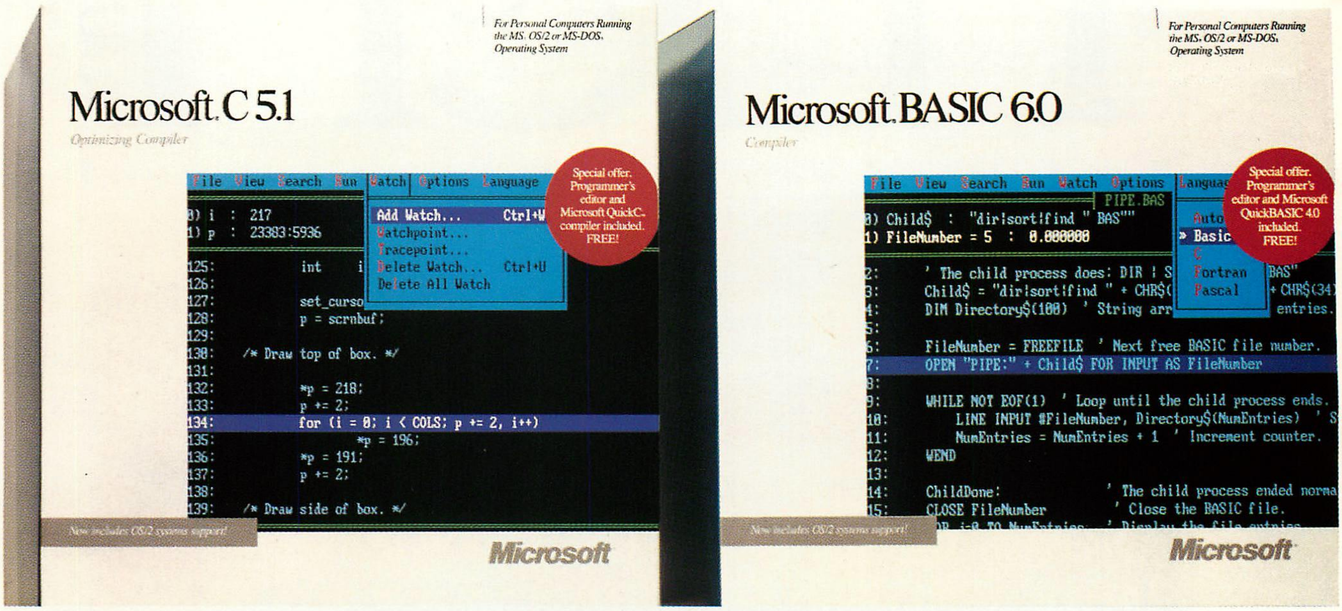


More specifically, Microsoft® QuickBASIC comes with on-line support: QB Advisor serves as a state-of-the-art electronic manual, and QB Express can teach you how to use the environment in just minutes. Of course, you'll be more productive even faster with our new Easy Menus and the instant environment.

Or maybe you already know how to program and want to be introduced to the power of C.

Microsoft QuickC® is your language. The fastest way to learn C: just use its integrated source-level debugger and advanced graphics capabilities. You'll be up and writing C programs in no time. Just like the name implies.

his own.



Professional.

Professional programmers are in a league all their own. So are the tools Microsoft provides the professional.

Each of our professional languages offers the complete development solution. Including the acclaimed CodeView.® An optimizing compiler for power. OS/2 support for access to the next generation of computing. And Microsoft Editor for the most sophisticated source code editing.

Our languages speak for themselves—all of these industry hard hitters use Microsoft languages.

As for the most powerful BASIC ever to grace a PC, there's BASIC Compiler.

With its talent you have the ability to create smaller, faster executables.

Which isn't to say Microsoft C Optimizing Compiler doesn't have a few talents of its own. Lightning fast at running executables, it also features an incremental linker and support for various powerful platforms like MS-DOS® and Microsoft Windows, as well as Presentation Manager.

So no matter what level of computer programmer you are, Microsoft makes a language that lets you be the best you can possibly be. Which goes to show you, one language company really does fit all.

Microsoft®
Making it all make sense.®

PageMaker is a registered trademark of Aldus Corporation. Lotus and 1-2-3 are registered trademarks and Metro is a trademark of Lotus Development Corporation. R:BASE is a registered trademark of Microrim, Inc. Oracle is a registered trademark of Oracle Corporation. WordPerfect is a registered trademark of WordPerfect Corporation.

If anyone tries to tell you that one language fits all, turn the page.

Our new MASM 5.1 gives you two things you've never had in a macro assembler.

MS OS/2 and spare time.


```
_DATA       SEGMENT     WORD PUBLIC 'DATA'
_DATA       ENDS
DGROUP      GROUP       _DATA
_TEXT       SEGMENT     WORD PUBLIC 'CODE'
            ASSUME      CS:_TEXT,DS:DGROUP,SS:DGROUP
            PUBLIC      _ShiftNsum
_ShiftNsum  PROC        NEAR
            push        bp                      ; setup stack frame with...
            mov         bp,sp
            sub         sp,2                    ;...one local variable [bp-2]
            push        CX                      ; save CX used during routine
            mov         WORD PTR [bp-2],0       ; clear local variable
            mov         Cx,WORD PTR [bp+4]      ; fetch number of sums
Sumlp:      mov         Ax,[BP+6]
            shl         Ax,cl                   ; double value in ax, cl times
            add         WORD PTR [bp-2],ax
            loop        Sumlp
            mov         Ax,WORD PTR [bp-2]      ; return value in ax
            pop         CX                      ; restore cx
            mov         sp,bp
            pop         bp                      ; restore stack and return
            ret
_ShiftNsum  endp
_TEXT       ends
            end
```

The old way: slow, cumbersome coding.

```
            .MODEL      SMALL,C
            .CODE
ShiftNsum   PROC        USES cx, Number:Word, sums:Word
            LOCAL       sumresult               ;declare a stack local
            mov         sumresult,0             ; clear sumresult
            mov         Cx,sums                 ; fetch number of sums
Sumlp:      mov         Ax,number
            shl         Ax,cl                   ; double value in ax cl times
            add         sumresult,ax
            loop        Sumlp
            mov         ax,sumresult            ; return value in ax
            ret
ShiftNsum   endp
            end
```

The new way: time off for good behavior.

As a professional programmer you know how fast MASM subroutines make your programs run. But you've also had to pay the price of cumbersome coding.

With Microsoft® Macro Assembler version 5.1, you get ease *and* speed. Starting with our *Mixed Language Programming Guide* that teaches you how to call MASM subroutines from your high-level programs. To the PROC directive that makes it as easy to pass parameters to an assembler subroutine as it is in high-level languages. Plus you get our easy-to-use "fill-in-the-blanks" template programs.

In addition to MS-DOS® support, MASM 5.1 has full MS® OS/2 support that lets you do things like break the 640K barrier, create multithreaded applications and Dynamic Link Libraries. And you get a complete set of high-octane development tools that lets you work faster under both environments. As well as the reconfigurable Microsoft Editor, a high-speed linker and new incremental linker for MS OS/2. And the renowned Microsoft CodeView® window-oriented source-level debugger that lets you blaze through debugging.

For the details, check the techbox. For more information, call (800) 541-1261, Dept. I17. And learn how today's new Microsoft MASM is so fast and easy it comes with the following offer.

The night off.

Microsoft Macro Assembler Version 5.1 for MS OS/2 and MS-DOS

Mixed-Language Programming Support and Documentation
- Mixed-Language Programming Guide with step-by-step examples.
- MASM templates for interfacing to BASIC, C, FORTRAN, and Pascal.
- Sample macros for interfacing for DOS and OS/2 systems. NEW!
- Simplified segment directives make writing MASM subroutines easy.
- Comprehensive instruction set *reference* with examples.
- Programmer's Guide – complete guide to MASM features and instruction set.

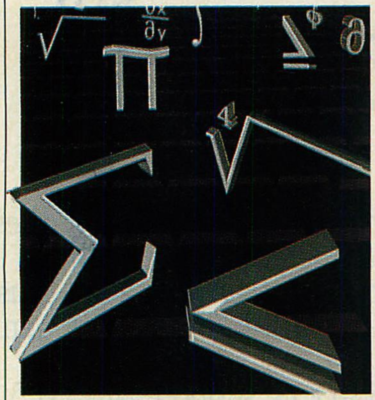
Macro Assembler for MS OS/2 and MS-DOS
- Full MS OS/2 support to break the 640K barrier. NEW!
- Supports the 80386 and 80387 instruction set.
- Assembles 25-40% faster than MASM 4.0.
- Have labels local to a procedure. NEW!
- Declare stack based local variables within a procedure. NEW!
- OS/2 systems incremental linker – up to 20 times faster than a full link. NEW!

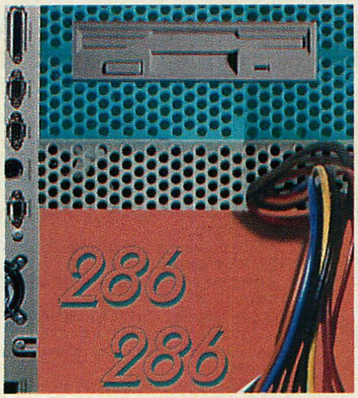
Microsoft CodeView Window-Oriented Debugger and Microsoft Editor
- Full OS/2 systems support. NEW!
- Debug MASM code using original source code with your comments.
- Interactively view structures and chase linked lists. NEW!
- Take advantage of the 80386.
- Set dynamic breakpoints, watch variables, memory, registers and flags.
- MS OS/2 and MS-DOS reconfigurable programmer's editor. NEW!

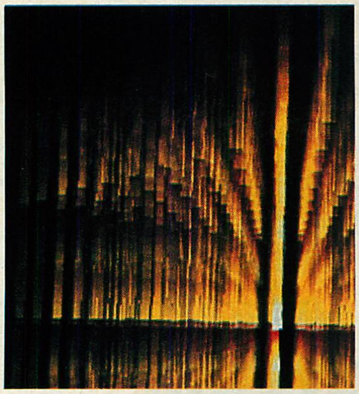
Microsoft®

PC TECH JOURNAL

FOR SYSTEMS DEVELOPERS AND INTEGRATORS







DEPARTMENTS

Cover illustration·Frank Riley

 CIRCLE NO. 236 ON READER SERVICE CARD

PC TECH JOURNAL

VOL. 6, NO. 12

PUBLISHER: *Donald J. Byrnes*
EDITORIAL DIRECTOR: *Will Fastie*
EDITOR: *Julie Anderson*

EDITORIAL
EXECUTIVE EDITORS: *Susan Holly, David Methvin*
SENIOR TECHNICAL EDITOR: *Jim Shields*
TECHNICAL EDITORS: *Maxine Fontana, Ted Mirecki, Chet Schuyler*
SENIOR EDITOR: *Gail Shaffer*
ASSOCIATE EDITORS: *David Robb, Jordene Zeimetz*
SENIOR COPY EDITOR: *Bruce Ansley*
COPY EDITOR: *Judith Estrin*
EDITORIAL PRODUCTION ASSISTANT: *Katherine M. Kiwak*
NEW PRODUCTS EDITOR: *Carole Eyring*
LAB TECHNICIAN: *Todd Bannar*
EDITORIAL ASSISTANT: *Barbara Smith*
BUSINESS MANAGER: *Trish Ledbetter*
EDITORIAL SECRETARY: *Beth Sauberlich*
RECEPTIONIST: *Tobie J. Ketterer*
CONTRIBUTING EDITORS: *Steven Armbrust, Dave Browning, David Claiborne, Peter C. Coffee, Richard M. Foard, Ted Forgeron, Philip N. Hisley, Thomas V. Hoffmann, Max Stul Oppenheimer, Kent Quirk*

ART AND PRODUCTION
ART DIRECTOR: *Sharon Reuter*
ASSOCIATE ART DIRECTOR: *Courtney Barone*
ASSISTANT ART DIRECTOR: *Crystal Hopkins*
ART ASSISTANT: *Cheri Glover*
PRODUCTION MANAGER: *Alison Regan Mrohs*
CONTRIBUTING ARTISTS: *Maciek Albrecht, David Povilaitis*

ADVERTISING SALES
NATIONAL ADVERTISING DIRECTOR: *Rita Burke*
NATIONAL TELEMARKETING MANAGER: *Arlene Braithwaite*
MARKETING DIRECTOR: *Marti Cunha*
ASSISTANT MANAGER OF MARKETING: *Kimberly Schroeder*
MARKETING COORDINATOR: *Debbie Lintner*
ASSISTANT TO THE PUBLISHER: *Lucy Frazer*
ADVERTISING COORDINATOR: *Mary Barrow*
DISTRICT MANAGERS: *Bill Barney–Southeast and Eastern Canada; Melinda Maginn–New England; John Blake–Mid-Atlantic; Stephen Dress–Midwest; Elaine Ebner, Deb Koch–West Coast*
ACCOUNT REPRESENTATIVES: *Patrick O'Brien–Southeast and New England; Nannette Vilushis–Midwest and Mid-Atlantic; Mary Martin–West Coast; Classified Advertising Director–Kathryn Cumberlander*

CIRCULATION
SUBSCRIPTION DIRECTOR: *Paul Stafford*
SINGLE-COPY SALES DIRECTOR: *Robert Woltersdorf*

GROUP VICE PRESIDENT: *J. Scott Briggs*

ZIFF-DAVIS PUBLISHING COMPANY, *a division of Ziff Communications Co.*
PRESIDENT: *Kenneth H. Koppel*
GROUP VICE PRESIDENTS: *J. Scott Briggs, William Lohse*
SENIOR VICE PRESIDENT, Marketing: *Paul H. Chook*
VICE PRESIDENT, Operations: *Baird Davis*
VICE PRESIDENT, Controller: *John Vlachos*
VICE PRESIDENT, Creative Services: *Herbert Stern*
VICE PRESIDENT, Research: *Marian O. White*
VICE PRESIDENT, Classified Advertising: *Alicia Marie Ivans*
VICE PRESIDENT, Circulation: *Bert Lacy*
VICE PRESIDENT, Circulation Services: *James Ramaley*
VICE PRESIDENT, Marketing Services: *Ann Pollak Adelman*
VICE PRESIDENT, Development: *Seth Alpert*
VICE PRESIDENT, Production: *Roger C. Herrmann*
DIRECTOR OF PLANNING: *Gary A. Gustafson*
PRODUCTION DIRECTOR: *Walter J. Terlecki*

ZIFF COMMUNICATIONS COMPANY
CHAIRMAN: *Philip B. Korsant;* PRESIDENT: *Kenneth H. Koppel;* EXECUTIVE VICE PRESIDENT: *Stephen C. Kahler;* SENIOR VICE PRESIDENTS: *Philip Sine, Hugh Tietjen;* VICE PRESIDENTS: *William L. Phillips, J. Malcolm Morris, Steven C. Feinman;* SECRETARY: *Bertram A. Abrams*

EDITORIAL OFFICE
PC Tech Journal, Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044-3506; 301/740-8300. FAX (group 3): 301/740-8809. MCI Mail: PCTECH. PCTECHline: 301/740-8383 (2400/1200/300 bps, 8 data bits, 1 stop bit, no parity). Telex: 6502565932 MCI.

ADVERTISING OFFICES
Corporate/Southeast/Canada: Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044-3506; 301/740-8300. *New England:* 144 Watching Avenue, Upper Montclair, NJ 07043; 201/509-8148. *Mid-Atlantic:* 89 Pearsall Place, Merrick, NY 11566; 516/546-7505. *Midwest:* Suite 1400, 180 N. Michigan Avenue, Chicago, IL 60601; 312/346-2600. *West Coast:* 3460 Wilshire Blvd., Los Angeles, CA 90010; 213/387-2100. 110 Marsh Drive, Foster City, CA 94404; 415/378-5580.

SUBSCRIPTION INQUIRIES
PC Tech Journal, P.O. Box 2968, Boulder, CO 80321. Subscription service: 303/447-9330. For back issues (subject to availability) send $7.00 per copy, $8.00 outside U.S. ($7.95/$9.95 for 1988 Directory issue only) to: Ziff-Davis Publishing Company, P.O. Box 5999, Cherry Hill, NJ 08034, Attn: F. Hunter; 609/354-4975.

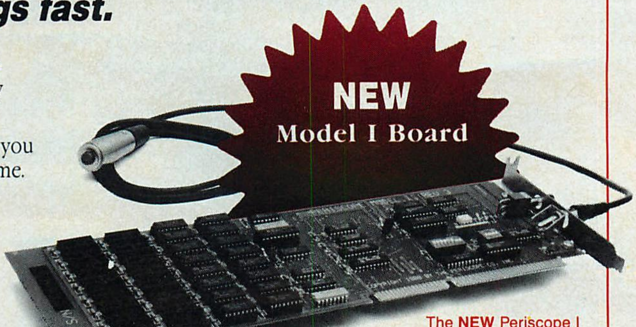
PC Tech Journal (ISSN 0738-0194) is published monthly by Ziff-Davis Publishing Company, a division of Ziff Communications Company, One Park Avenue, New York, NY 10016. Subscription rate is $34.97 for one year (12 issues). Additional postage for Canada and Foreign is $10.00 per year. Second-class postage paid at New York, NY, and at additional mailing offices. POSTMASTER: Send address changes to *PC Tech Journal*, P.O. Box 55761, Boulder, CO 80321.

Periscope's *New* Version 4

...Gives you all the right stuff for debugging! No matter which model you pick, you have the same powerful software to help you track down hard-to-find bugs fast.

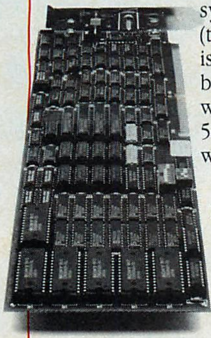
Periscope's hardware adds the power to solve the really tough debugging problems.

The break-out switch lets you break into the system any time. You can track down a bug instantly, or just check what's going on, without having to reboot or power down and back up. That's really useful when your system hangs! The switch is included with Periscope I, Periscope II, and Periscope III.



The **NEW** Periscope I memory board keeps all debugging information out of the lower 640K. Can be used in PCs, ATs, and 386s with both EGA/VGA and EMS boards installed. The Periscope break-out switch enables you to recover from a hung system. Included with Models I, II, and III.

Periscope I has a **NEW** board with 512K of write-protected RAM, user-expandable to 1MB, for the Periscope software, symbol tables, and all related debugging information. Normal DOS memory (the lower 640K) is thus totally freed up for your application, and Periscope is protected from being overwritten by a run-away program. The new board's footprint is only 32K, so you can use it in PC, AT, and 386 systems with EGA/VGA and EMS boards installed (not possible with the previous 56K board). It can also be used with Periscope III to provide additional write-protected memory.



Periscope III has a board with 64K of write-protected RAM to store the Periscope software and as much additional information as will fit. AND...

The Periscope III board adds another powerful dimension to your debugging. Its hardware breakpoints and real-time trace buffer let you track down bugs that a software-oriented debugger would take too long to find, or can't find at all!

The Periscope III hardware-breakpoint board captures information in real-time, so you'll find bugs that can't be found with a software-based debugger.

Periscope's software is solid, comprehensive, and flexible.

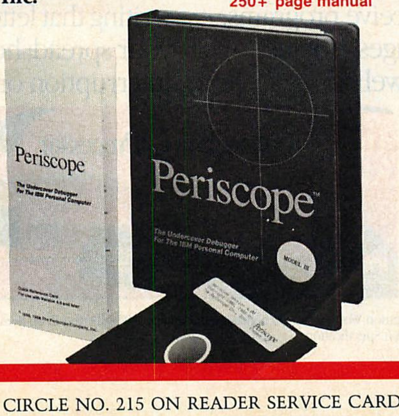
It helps you debug just about any kind of program you can write... thoroughly and efficiently.

Periscope's the answer for debugging device-drivers, memory-resident, non-DOS, and interrupt-driven programs. Periscope works with any language, and provides source and/or symbol support for programs written in high-level languages and assembler.

David Nanian, President of Underware, Inc. (of BRIEF fame) says this about the new Periscope Version 4:

"Periscope has always been an unbelievable assembler-level debugger. Version 4 has turned it into a terrific source-level debugger as well. Aside from major enhancements like the source-level improvements, all the little changes make a really big difference, too. For instance, symbol lookups and disassemblies are noticeably faster, and highlighting the registers that have changed really makes life easier. Once again, Periscope has raised the industry standard for debuggers!"

Periscope software & 250+ page manual



What's New in Periscope Version 4:

- View local symbols from Microsoft C (Version 5)
- Debug Microsoft windows applications
- Set breakpoints in PLINK overlays
- Improved source-level support
- Monitor variables in a Watch window
- 80386 debug register support
- Debug using a dumb terminal
- PS/2 watchdog timer support
- Use mixed-case symbols
- Set breakpoints on values of Flags
- Much more!

- **Periscope I** includes a **NEW** full-length board with 512K of write-protected RAM; (user-expandable to 1MB); break-out switch; software and manual for $695.
- **Periscope II** includes break-out switch; software and manual for $175.
- **Periscope II-X** includes software and manual (no hardware) for $145.
- **Periscope III** includes a full-length board with 64K of write-protected RAM, hardware breakpoints and real-time trace buffer; break-out switch; software and manual. Periscope III for machines running up to 10 MHz with one wait-state is $1395.

Due to the volatility of RAM costs, prices on board models are subject to change without notice.

REQUIREMENTS: IBM PC, XT, AT, PS/2, 80386 or close compatible (Periscope III requires hardware as well as software compatibility, thus will not work on PS/2 or 80386 systems); DOS 2.0 or later; 64K available memory (128K at installation time); one disk drive; an 80-column monitor.

Call us with your questions. We'll be happy to send you free information or help you decide on the model that best fits your needs.

Order Your Periscope, Toll-Free, Today!
800-722-7006

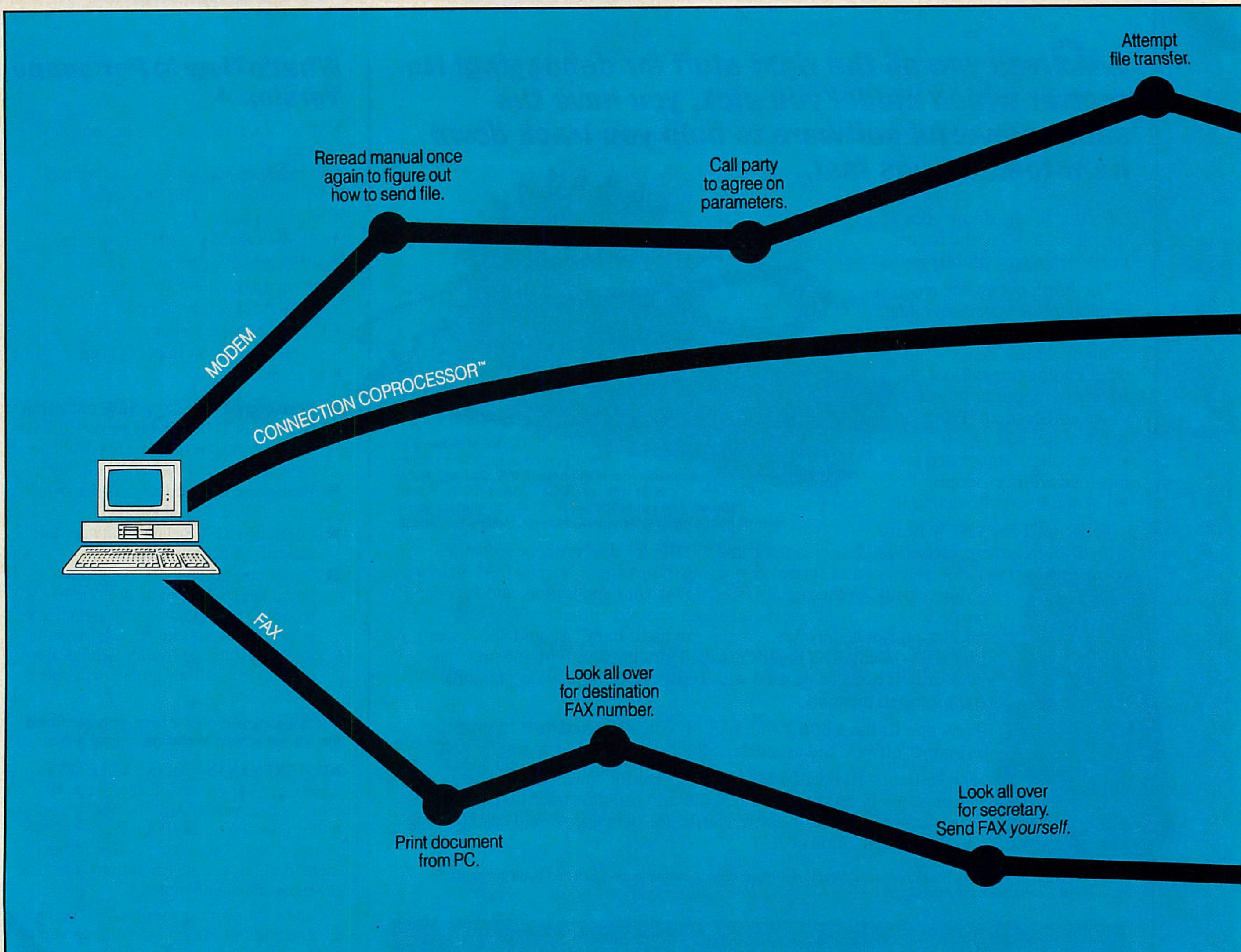
MAJOR CREDIT CARDS ACCEPTED

The Periscope Company, Inc.

1197 PEACHTREE ST. • PLAZA LEVEL
ATLANTA, GA 30361 • 404 / 875-8080

CIRCLE NO. 215 ON READER SERVICE CARD

Intel announces



Announcing the first non-stop communications route between businesses anywhere in the world.

Intel's Connection CoProcessor.™

It's a whole new way to send and receive programs, files, text, graphics and electronic messages. Not just between PCs, but with fax machines as well.

The Intel Connection is a new communications coprocessing board that frees your PC from the task of communicating.

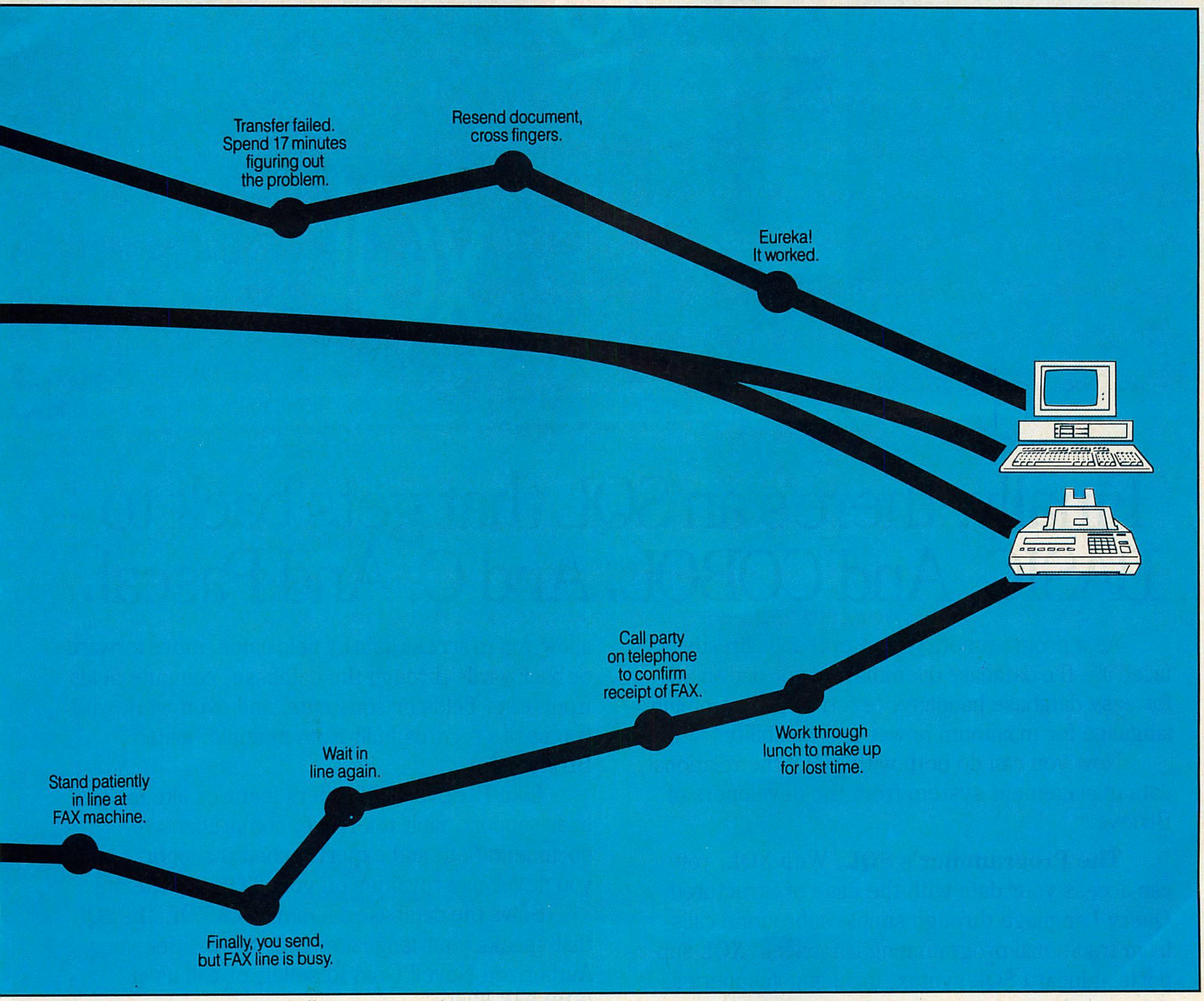
Which means you can work on your computer non-stop—even while sending a document as big as the New York phone book.

Now your faxes or files can fly cross country. And you can still be zooming around in your flight simulator. Or writing that letter to your biggest account. Or recalculating your spreadsheets for tomorrow's meeting. Without interruption or downtime.

And when you fax, there are no more stopovers at the printer, or cooling your heels in line at the fax machine. Because Connection lets you do everything without leaving your PC.

And when you send files directly to

non-stop flights.



another Connection-equipped PC, you won't have to fiddle around with modem parameters or protocols. Because the Connection CoProcessor sets them for you. At 9600 bps.

Best of all, Connection supports CAS,* a new communication standard that is supported by major software and hardware developers like Symantec,™ Borland,® DCA,® Microsoft,® and WordPerfect.® Which means you can send, for example, a word processing file to your London office, without ever exiting your WordPerfect program.

You just send the document directly from your application. That's it. There isn't any complicated communications software to learn.

And everything is backed by toll-free technical support and a five-year warranty from Intel.

So what are you waiting for? Call 800-538-3373 now for a free demo disk.

Because with Intel's Connection CoProcessor, communication is really going to take off.

intel®

Corporation. DCA is a registered trademark of Digital Communications Associates, Inc. Microsoft is a registered

CIRCLE NO. 229 ON READER SERVICE CARD

Finally, there's an SQL that gets back to BASIC. And COBOL. And C. And Pascal.

As a programmer, you've probably already faced it—the database dilemma. Do you use an SQL for easy database handling, or a true programming language for maximum power and flexibility?

Now you can do both with XQL,® the relational data management system from the developers of Btrieve.®

The Programmer's SQL. With XQL, you can access your data with the ease of Structured Query Language through simple subroutine calls from traditional programming languages. XQL supports standard SQL syntax, including subqueries, unions and security groups.

XQL Relational Primitive Operations. In addition, XQL lets you bypass the SQL level and perform highly efficient, relational primitive operations directly. You get all the functionality of a relational database model without the constraints of a 4th generation language.

Building on Btrieve. The heart of Novell's family of data management tools is Btrieve. By letting you access multiple records at a time, XQL adds a powerful dimension to Btrieve. XQL incorporates sophisticated data manipulation features which allow you to access data by field name, move forward or backwards through the database, compute fields from other fields or constants, and even work with composite records built from multiple, joined Btrieve files.

Like Btrieve, XQL offers features like multi-user support, fault tolerance, comprehensive documentation, and expert technical support. And you never pay royalties on your XQL applications.

Solve the database dilemma with XQL, the SQL that speaks your language. Only $795.* See your Authorized Novell Gold Reseller, or call us at (512) 346-8380.

For more information, call from your modem 1-800-444-4472 (8 bit, no parity, 1 stop bit) and enter the access code NVXQL3.

NOVELL

For software solutions, you should be seeing red.

*Suggested retail price (US dollars) ©1988 Novell Inc., World Headquarters, 122 East 1700 South, Provo, Utah 84601 (801) 379-5900
Requires Btrieve 4.x and PC-DOS or MS-DOS 2.x, 3.x.

CIRCLE NO. 201 ON READER SERVICE CARD

Systems Perspective

Call for a Truce

Unix and OS/2 are both worthy of winning the operating-system battle. The victor is best determined by examining the target market.

A religious war is raging over whether Unix or OS/2 is the better operating system. The OS/2 sect believes that Unix is not a serious operating system; that its non-mnemonic commands, such as **awk** and **grep**, are difficult to understand; and that few horizontal applications exist for it.

Unix followers swear that Unix is a mature operating system worthy of a developer's respect and that OS/2 is merely a single-user Unix with mnemonic commands. In the midst of these slings and arrows, I raise a white flag and call for a truce.

Unix and OS/2 differ in some fundamental philosophies, but at the API level, they are roughly equivalent. Both provide multitasking, interprocess communications, and virtual memory management. The principles are the same, although each operating system implements them in a slightly different way.

Presuming, then, that both Unix and OS/2 are acceptable development environments, how does a developer choose between them?

The choice depends less on ease of development and more on the developer's target market. This has certainly proven true for DOS. Developers have leaped technological walls, working long hours to squeeze code into 640KB, so their applications can run in the largest installed base.

Today, however, developers stand at a crossroad. Having run up against roadblocks in DOS that they cannot hurdle, developers must choose an alternate route.

The critical questions to answer in selecting a new route are: How do you want to manage multiple users? What user interface do you need? Are you automating a desktop or delivering a vertical application? What applications must run hand in hand?

The answer to the first question is clear-cut. Unix is designed to be inherently multiuser, meaning that multiple users can share one processor through dumb terminals. This mainframe- and minicomputer-like model offers significant cost savings over buying a computer for each user, especially as the number of users increases. OS/2, on the other hand, is a single-user system at the processor level. Multiple users are handled by hooking up multiple PCs to a LAN; this way, users can easily share applications, data, and peripherals while maintaining some private applications, data, and peripherals.

The question of user interface is still a little muddy. Both sides have great expectations for their graphics user interfaces, but so far, little of these interfaces has materialized. The ultimate success of Unix's X Window System and OS/2's Presentation Manager is yet to be seen. In the meantime, Unix does offer programmable script languages that can be used to create a custom text-based interface for the user to select and execute applications.

Last but not least are two questions that deal with the type of application being developed. Most existing Unix applications are vertical business applications (such as accounting, order entry, and payroll), while most promised OS/2 applications will automate the desktop (such as spreadsheets, word processors, and so-called groupware). This is merely how the markets have developed; there is no inherent reason in either operating system that limits its applications to a particular market segment. Horizontal applications exist for Unix, and developers are creating vertical applications for OS/2.

To choose the best operating system for the job means closely examining the target market. For an internal, mission-critical application that runs by itself, either operating system is a viable option. If, however, you are going to sell the application into the general-purpose market, you should carefully examine the availability of needed auxiliary applications.

Neither Unix nor OS/2 is an intrinsically realtime operating system. Each implements a fairness scheduling algorithm, and OS/2 is optimized for interactive response time. We plan to look at the current crop of realtime operating systems in a future issue.

PEACE TREATY

Given that Unix is a capable operating system for many different kinds of applications, the question becomes: Why is OS/2 gaining any ground at all? Theoretically, OS/2 gives DOS developers and users a common flag to rally

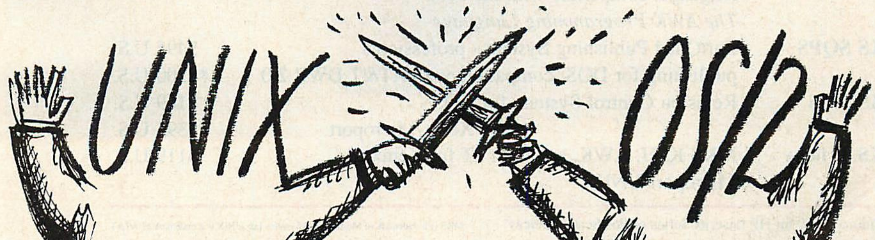


ILLUSTRATION • MACIEK ALBRECHT

around, a unified target market to move toward. Collective wisdom says developing applications for OS/2 should be a safe bet because OS/2 will eventually replace the vast DOS market.

Eventually is the key word. In the meantime, Unix is ahead of OS/2 in exploiting the 386 and in availability of development tools. Not wanting to miss any sizable market, many developers are moving their applications first to Unix and then to OS/2. It isn't really a question, then, of which operating system to choose; rather, it is a question of which operating system to begin development on first.

Our cover suite this month explores the worlds of Unix and OS/2. In "Worlds Apart, Worlds Together" (page 50), Robert R. Morris and William E. Brooks present both the historical and market contexts of each operating system. Morris and Brooks then examine the APIs of each operating system in "At the Core: An API Comparison" (page 62), giving valuable insight into writing code that ports easily between Unix and OS/2.

Although developers may face a more difficult time in maintaining multiple operating system versions of their applications, integrators now have it made. With developers creating applications to run on both Unix and OS/2, integrators are free to choose appropriate platforms for each group of end users they support.

If Unix is appropriate, integrators know they can choose it with little fear of locking themselves out of future horizontal applications. They also can integrate Unix, OS/2, and DOS platforms for sharing data or peripherals—without paying a penalty for choosing what they might have previously considered the wrong platform.

DISTINGUISHED DESIGN

We are honored to have won the Ozzie Award for Design Excellence, sponsored by *Magazine Design and Production*. From more than 1,100 entries, our June 1988 issue was selected to receive the bronze award for Best Overall Design of a trade publication with a circulation of more than 100,000.

I am pleased to see this well-deserved award extended to art director Sharon Reuter and her staff, Courtney Barone, Crystal Hopkins, and Cheri Glover. They work hard each month to create art that enhances the message of our abstract subject matter, making it more inviting to the reader. Congratulations to our art department.

What turns four useless, disconnected monochrome monitors and keyboards into four speedy, graphic workstations

for about the cost of an executive desk?

"Major Breakthrough Makes Programmers an Average of 106% More Productive"

Remarkable new tool is so powerful, it's guaranteed to double your productivity and pay for itself in less than two weeks, or it costs you nothing.

by Bruce Lynch, President of Solution Systems, Inc., specialists in boosting programmer productivity

South Weymouth, MASS.—Today I bring you news of a tool so powerful, it's boosting programmer productivity an average of 106%. That's the extraordinary finding of a survey of some 2,000 programmers who've tried Brief, the revolutionary PC Program Editor now available in a new version 2.1.

The scientific survey (the results of which are available on request) was conducted among programmers in all languages, including Assembler, Basic, C, Cobol, dBASE, FORTRAN and Pascal, among others.

"But It's Unthinkable . . . Uncomfortable"

Of course, for many programmers, switching to a new program editor is "unthinkable right now . . . I'm just too busy to learn a new editor . . . or too 'comfortable' with what I've got."

If you're in this "rather-fight-than-switch" group, *Brief was designed for you.* It's so easy to master that you're up-and-running in 15 minutes . . . effortlessly surpassing your previous editor's performance in one hour. After that, you're flying.

In fact, Brief is *guaranteed* (or your money back in full) to double your output . . . be the easiest-to-use program editor you've ever encountered . . . literally pay for its purchase price in less than two weeks . . . boost your creativity into a higher gear . . . and add a whole new dimension of excitement and ease to your programming.

Hard to believe? Well, more than 40,000 copies of Brief have been sold, all with a complete money-back guarantee. *Less than 1% have been returned.* Perhaps even more impressive is what industry leaders are saying:

"Revolutionary"

"Exudes power. Programmers feeling limited by more basic editors will look upon Brief as a liberation. . . ."—*Billy Rubin, The C Gazette*

"So far surpasses users' expectations that it may fairly be called revolutionary."—*John Cowan, MicroTimes Magazine*

"The best programmer's text editor I've ever seen . . . *a tour de force*."—*William Zachmann, Computerworld*

"If you need a general-purpose PC programming editor, look no further."—*Jerry Pournelle, Byte*

"I used to program with WordStar. But as soon as I saw Brief, I jumped into it."—*Wayne Ratliff, designer of dBASE*

"Simply the best text editor you can buy."—*John Dvorak, Infoworld*

"Staggering" List of Features

"The best code editor I have seen....Brief is to editors as Lotus 1-2-3 is to spreadsheets.... Its list of features is staggering."—*David Irwin, Data Based Advisor.* For example:

Unmatched Power and Speed. Brief is unconditionally guaranteed to make you a *faster programmer* than you can be with any other editor, bar none. It's loaded with every imaginable feature to streamline your programming. Such as:

Reconfigurable. You can customize Brief any way you want—even make it behave exactly like a much more powerful version of your *present* editor. *Computerworld* raves, "Brief has an extremely powerful macro language that can be used to create new commands, modify existing commands, tailor the program to any programming language and create an initial setup to suit the user in almost any situation."

You can refigure your keyboard to assign a command with any character sequence—so you won't have to learn a new keyboard. If there's *anything* you don't like about Brief, fine—*change it easily,* or customize new features you've always dreamed of having.

Breakthrough "UNDO" (Not Just "UNDELETE"). "Brief's UNDO feature," says *Microtimes,* "is astounding in it scope and power. . . . Having the ability to undo the mistaken deletion of a whole page of valuable information is very reassuring."

There is *no* command affecting your text that cannot be undone. Say you send your editor on a global change, and realize you've made a mistake. UNDO brings you back to where you started. Incredibly, you can undo *your last 300 commands or keystrokes.* You'll save hours and repair mistakes so much faster. Best of all, you'll be free to explore interesting branches of code without fear. A profound liberation in your creativity.

"The UNDO capability is easily the best of all editors tested."—*PC Tech Journal*

"UNDO doesn't just recover accidental deletions, it undoes accidents. . . ."—*Byte*

"Can save endless grief."—*Dr. Dobb's Journal*

"I almost look forward to making mistakes."—*The C Journal.*

Multiple Windows. Brief gives you unlimited file size and an unlimited number of simultaneous files. You can load all the files related to your project at one time . . . then split the screen into as many windows as you need . . . then size each window to see just the section you want. You can easily move text between windows, even edit the same file in several windows with a single command.

Immediate Compilation. Run any compiler—even the largest—without having to stop editing. Never again waste time reloading your editor. Brief will call your compiler, then place you at the line of an error—or even a potential error—with one keystroke.

Template Editing. Enter abbreviations and Brief expands them into entire constructs.

Expressive Searching goes beyond "wildcards" with full UNIX-style regular expressions.

"Escape to DOS" (while inside Brief) saves you countless hours of reloading.

Full Technical Support, always free.

Plus scores of other features. . . . And it's not copy protected.

1-Hour Trial (Guaranteed 100-to-1 Return)

Try Brief for just one hour. If you're not convinced it will save you at least 100 hours of tedium in the coming months (giving you a 100-to-1 return on the hour you invest with it) . . . if it doesn't also prove it has the power to boost your productivity by as much as 106% or more . . . be the easiest-use program editor you've ever encountered . . . pay for itself in less than two weeks . . . shift your creativity into a higher gear . . . and add a new dimension of excitement to your programming . . . *then just send it back at any time within 90 days of your purchase and get a 100% refund.*

FREE If You Order Immediately...a disk of the 65 most powerful macros ever designed for the world's #1 program editor.

How to Try Brief Without Risk

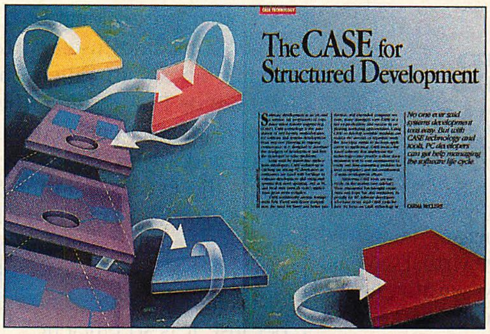
Brief costs only $195 (plus $4 shipping and handling). As *Computerworld* says, "it's worth every penny." Best of all, you can try it *risk-free* thanks to the 90-day, 100% money-back guarantee. To try Brief without risk and receive your free bonus, just call toll-free:

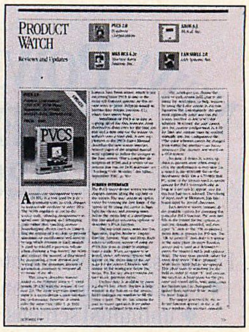
800-852-8523, ext. 202 PCTJ 12/88

Solution Systems, 541 Main Street
Suite 410, South Weymouth, MA 02190

All major credit cards . . . Volume discounts.

LETTERS





EVOLUTION OF A METHOD

I recently read the article, "The CASE for Structured Development" (Carma McClure, August 1988, p. 50). As someone working in the technological field of computer-aided systems engineering (CASE), I would like to add some points and correct some omissions in the area of my own specialization—the Jackson methodologies.

First, I applaud the general thesis of McClure's article—that CASE tools and methods are inseparable. She does an admirable job in classifying the various methods available and outlining the main features of each.

In the section on the Jackson methods, however, I was disappointed to discover that McClure confined her discussion to Jackson Structured Programming (JSP), which is referred to in the article as the Jackson Design Methodology, the name by which JSP used to be known.

JSP is a program-design method; its starting point is a program specification, and it takes the designer through steps that result in a structured design for the program code. However, JSP is not the only Jackson method. Since 1981, we have been expanding a full development method called Jackson System Development (JSD), which covers a much larger part of the development life cycle.

JSD is a software engineering method that addresses activities in the analysis, design, and programming phases and is applicable to both data processing and realtime systems. Like other methods, it makes a strong separation between logical and physical design; unlike others, however, it uses composition rather than decomposition as the basic technique in constructing the specification.

Using JSD, a full specification is put together in two main stages. The first stage involves identifying the entities with which the system is concerned and describing these entities in terms of both their data and their dynamic behavior and interaction. The second stage involves adding definitions of application functionality to the entity model.

Particular strengths of the JSD method are that the same concepts and notations are used both for the entity descriptions and for the functionality; these descriptions are executable, making the early prototyping of the specification possible.

The software support available for JSD enables specifications to be entered and manipulated in graphics form and automates the production of prototypes and production code from these specifications.

Returning to the JSP method, I didn't altogether agree with the remarks at the end of the summary (which was otherwise an accurate account of the JSP method). McClure states that JSP is weak in adding logic to control iterations and conditionals. In fact, with its technique of backtracking and multiple read-ahead, the JSP method has more to say about, and provides more help with, this aspect of logic design than other well-known CASE methods.

In addition, McClure makes a remark to the effect that JSP does not directly apply to complex programs. While it is true that the JSD method is more effective than JSP in modularizing systems, JSP is widely used on programs of all sizes with great success.

I hope that it may be possible for *PC Tech Journal* to devote some space in a future issue to a fuller review of the Jackson methods—particularly JSD. Although they are well known and widely used in Europe, we are only just beginning to market the Jackson methods on this side of the Atlantic.

A. T. McNeile, principal consultant
Jackson Systems Corporation
Windsor, CT

Thank you for the update on how the Jackson Structured Methodology has evolved to address multiple phases of the life cycle. All of the major structured development methodologies that were introduced in the 1970s have evolved to better meet user needs, to include newer technologies such as database design, and to cover the development of more types of systems—small and large, realtime and on-line transaction systems. I expect them to change further to take advantage of newer life cycle models such as prototyping, analysis techniques such as Joint Application Design (JAD), and CASE and software automation.

The purpose of my survey article was to introduce the fundamental concepts underlying each methodology and to provide an overview of their basic steps. In the space available, it was impossible to thoroughly explain and critique multiple methodologies.

—Carma McClure

Judging by the number of responses generated by our cover suite, CASE is growing in popularity with application developers. We will follow up on our CASE coverage in future issues.

—MF

PVCS PUT STRAIGHT

I wish to express my appreciation for the comprehensive article on configuration management ("Tracking Code Modules," Jim Vallino, September 1987, p. 50) and the thorough review of the Polytron Version Control System (PVCS) version 2.0 (Product Watch, Jim Vallino, October 1988, p. 131). I would like to point out some clarifications and corrections, however, with respect to the review of PVCS 2.0.

First, Mr. Vallino accurately notes the relatively slow performance of the optional screen interface on a 4.77-MHz IBM PC with a CGA. According to our research, over half of our customers

and other professional programmers in larger companies use 8-MHz or faster AT machines with either a monochrome or EGA display. The difference in performance between the test machine and a typical machine is significant—the PVCS screen interface is more than five times faster in a typical environment. In addition, over 80 percent of our customers use the command-line interface almost exclusively because it is so efficient.

Second, the review points out that the screen interface does not automatically spawn a new DOS shell. It is possible, of course, to do so manually by executing Command. More important, however, is that our method allows you to invoke some other shell, such as Polytron's PolyShell, instead of DOS's COMMAND.COM, when you want a new shell invocation. Mr. Vallino also points out the symptom of a latent bug in PVCS version 2.0 with regard to the use of aliases. The current release, 2.1, does not have this problem.

Finally, the interpretation of the use of the message-file capability is not quite correct. This is an extremely powerful feature that immensely simplifies the accurate documentation of changes to large numbers of files. If you use the −M@file option to the **put** command, the change description is taken from the named file. However, if the file name is omitted, as in −M@, the name of the message file is computed by doing a suffix translation on the name of the file being **put**.

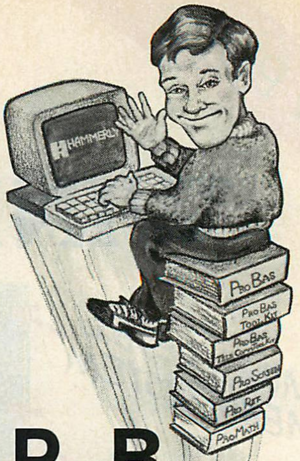
Donald K. Kinzer
Polytron Corporation
Beaverton, OR

DATA STRUCTURE DEBATE

I must respond to Evan P. Provisor's letter ("Free the Data Structure," September 1988, p. 15) concerning the publication of data structures. First, to state that a file structure "does not provide much usable information as to the inner workings of the program code" is ridiculous. Perhaps this is true with a simple flat ASCII file, but for such a file one would hardly need a set of data structures. If the file contains pointers and other such control information, these pointers could well be the key to the software's performance.

Second, it is curious how Mr. Provisor supports his point about developers not losing business by publishing their structures. He cites his frustrations with converting a customer from one system to another, then chides the losing vendor for not pro-

~~Five~~ Six easy ways to boost your BASIC



ProBas™

updated

Professional Basic Programming Library

Announcing **ProBas** Version 3.0, now with over 365 assembly routines to really kick QuickBASIC and BASCOM into high gear. BYTE magazine calls **ProBas** a "Supercharger for QuickBASIC". Thousands of programmers rely on **ProBas** to make their life easier and to enhance their programs with features like:

- **A 1,000 page 2 volume manual**
- **Full-featured windowing**
- **Screen snapshots (Text & Graphics)**
- **String, array, and pointer sorts**
- **Lightning-fast file I/O**
- **Full mouse support**

Create dazzling screens in text mode, CGA, EGA, VGA or Hercules graphic modes. Save and restore screen snapshots to arrays, EMS memory or files. Full featured windowing to meet the most demanding jobs. The **ProBas** system of virtual screens allows you to draw full or partial screens to memory, and then snap them on in an eyeblink. You can even create vitural screens far larger than the display screen.

Sick of running out of string space? Store hundreds of K in numeric arrays or megabytes in extended or expanded memory. Tired of using a kludgy SHELL to DIR to read a directory or archive files? Scan subdirectories or .ARC files using wild-cards and store thousands of file names, dates, and times. Wish you could drag a window containing text or a menu around the screen with a mouse? It's easy!

ProBas gives you a complete set of blazingly-fast file routines. Read or write huge chunks of data at a clip, with file locking and error handling so that you can even use them in subprograms. You'll never want to use BASIC's file I/O again! Sort data with lightning fast array and pointer sorts. Search files or arrays at assembly speeds. **ProBas** also has over 200 other essential services including handy string, date, time, directory and array manipulation, string, screen and data compression, full mouse support, valuable equipment and input routines and faster replacements for most BASIC commands.

Whether you are a professional or a novice, **ProBas** will boost your BASIC in ways you never dreamt possible. **ProBas** allows professionals to save time and work and lets novices write professional-quality programs quickly and easily. After all, how much is a few hundred hours of your time really worth?

For all versions of QuickBASIC and BASCOM including BASCOM 6.0 for OS/2. ***Just $135.00!***

ProRef™

On-Line Help For ProBas

ProRef provides pop-up help for the routines in **ProBas** and is an extension of the QuickBASIC programming environment. Find help on any routine with a few keystrokes or mouse clicks. Pop-up an ASCII chart, calculator, scan code module, box diagram, your own help information or almost any DOS program via a hot-key. ***Just $50.00!***

ProScreen™

Professional Screen Management System

ProScreen is a full-featured screen generator/editor that will save you more design and coding time than you ever thought possible. **ProScreen** treats screens like a word processor treats text to provide complete control over characters, colors, and placement. Design input screens with up to 130 fields and 19 pre-defined and 2 user-defined masks. Use **ProBas** or the included BASIC/Assembler subroutines to access the screens. No kludgy code generators here! Comes with subroutine source, extensive on-line help, and a 285 page manual. ***Just $99.00!***

ProMath™

new!

ProMath is a collection of over 150 high-level routines that provide mathematical functions and operations for programmers who often work in mathematics, science, or engineering. Complex variables, real and complex matrices, real and complex trigonometric and hyperbolic functions and their inverses, solution of linear equations, integration, differential equations, Fast Fourier transforms and many other useful routines are provided.

For years Fortran has been the language of choice for scientific and engineering applications, but it lacks many of the useful features of QuickBASIC. **ProMath** contains most of the Fortran mathematical and numeric functions and allows you to easily translate Fortran code to BASIC or write new programs in BASIC while retaining Fortran's numerical prowess.

The **ProMath** manual is over 200 pages and provides a complete description of each routine, including any algorithm and the mathematical formula the routine uses, shown in standard notation. For QuickBASIC 4 and BASCOM 6 only. ***Just $99.00!***

ProBas™ ToolKit

The **ToolKit** is a collection of assembly and BASIC modules that use the **ProBas** library to save you even more hours of grunt work. Why spend hundreds of hours re-inventing the wheel when you can just plug in **ToolKit** modules like:

- **Menu Generators**
- **Fast B-tree indexing**
- **Mini-editor with word-wrap**
- **Patch .EXE files**
- **Protected storage areas**
- **Julian date routines**

The **ToolKit** also includes clock, calendar, windowing, BCD math routines and much more, complete with BASIC source code and a comprehensive manual. The **ProBas ToolKit** adds capabilities and helps conserve your most valuable asset of all—time! Requires **ProBas**. ***Just $99.00!***

ProBas™ TeleComm ToolKit

The **ProBas TeleComm ToolKit** is a collection of high-level communications modules that you plug into your code to provide popular file transfer protocols, terminal emulations, login scripts and baud rates up to 115,200 baud. You get:

- **Xmodem/Modem7/Xmodem-1k**
- **Ymodem (single and batch)**
- **CRC-16 and Checksum**
- **VT52, VT100, ANSI BBS etc.**
- **Auto Dialer & data base**
- **Documented BASIC source**

Why use clumsy SHELLs to complex terminal programs when you can plug just the communications routines you need into your code? Implement just the features and commands you want. Requires **ProBas**. ***Just $75.00!***

Our thirty day, money-back guarantee assures you the highest quality and our technical support staff is always ready to help. Try our BBS at **(301) 953-7738** or give us a call at:

HAMMERLY COMPUTER SERVICES, INC.

8008 SANDY SPRING ROAD • LAUREL, MD 20707

(800) 343-7484

INT'L. ORDERS: (301) 953-2191 FAX: (301) 725-8147

Add $5.00 per item ($8.00 Canada) for shipping per order. Europe: $39.00 for 1st item plus $5.00 for each additional item. Visa, M/C, C.O.D. (US Only) checks and approved POs accepted. Trademarks **ProBas**, **ProRef**, **ProScreen** **ProMath**: Hammerly Computer Services, Inc. QuickBASIC, BASCOM: Microsoft Corp.

Please circle 139 on reader service card.

viding data structures. Apparently, Mr. Provisor had no such problems with the second vendor. If vendors could be sure they would be on the winning side of the deal, they might be willing to provide the necessary information.

The most serious issue here is one of control. As developers of school-administration software, we are barraged with eager requests for data structures, which we must decline. The problem is that once someone starts monkeying around with our data files directly, we lose all the control we have built into the system to guarantee the integrity of the data. I could guarantee that within weeks of publishing our data structures, we would be flooded with phone calls from people wondering why the data are all messed up, and blaming it on us!

Data structures are legitimately proprietary. They are usually part of the copyrighted source code. To open them up would be an invitation to circumvent the software's stringent security and integrity checks. These are all excellent reasons for developers not to publish their data structures.

Randall Stokes
Olympia Computing Company Inc.
Tumwater, WA

GRAVITY'S FALL

The drawing accompanying Julie Anderson's column in the September 1988 issue ("Measuring Performance," Systems Perspective, p. 9) implies that an object dropped from a window falls at a constant velocity of 9.81 meters per second (m/s). The figure should read 9.81 m/s/s to take into account the acceleration due to the earth's gravity (neglecting air friction).

David V. Fansler
Roche Biomedical Laboratories Inc.
Burlington, NC

COMMENTS WELCOME

All letters to the editor should be directed to Editor, *PC Tech Journal*, Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044. Correspondence also can be submitted over MCI Mail to PCTECH.

Although *PC Tech Journal* cannot publish all letters received, every effort is made to answer as many as possible. Please keep letters to the point, and include name, mailing address, and telephone number; when a letter is lengthy, a diskette is appreciated.

NEW DIRECTIONS

EISA: A Mistake

A vendor consortium complicates matters with a third bus.
□ Also, new machines from Apple. □ Systems Forum '89.

The Extended Industry Standard Architecture (EISA) is a mistake. That is the consensus opinion of *PC Tech Journal*'s editorial staff, which has been poring over the limited EISA documentation and talking with vendor companies, including IBM, since a coalition group headed by Compaq announced EISA in September. We do not come to this opinion lightly, but we are firm in the belief that EISA is an unnecessary complication in the PC industry and one that does not benefit the end user.

Sadly, political and emotional issues dominate the EISA matter; the EISA consortium (especially Compaq) is using EISA to focus attention away from IBM just when the PS/2 family may be picking up steam.

To get to the bottom line, IBM's Micro Channel architecture offers a well-designed bus that debuted most of the features of EISA. Moreover, it is here today and can therefore be exploited today. The Micro Channel is a superior technical solution by at least a small margin. The industry will better serve the interests of those buying the next 50 million desktop computers if it migrates to the Micro Channel instead of promoting yet another architecture.

DEFINING EISA

Providing a technical description of EISA is almost impossible. Unlike IBM, which provided substantial technical detail at the time it announced the PS/2, the Compaq-led EISA consortium (which initially included AST, Epson, Hewlett-Packard, NEC, Olivetti, Tandy, Wyse, and Zenith, but now counts just about everybody except IBM) distributed only a sketchy technical synopsis that describes concepts but has no meat. The actual specification is available only to developers willing to sign a nondisclosure agreement and to pay an administrative fee of $2,500 (interested parties should contact the law firm of Bishop, Cook, Purcell & Reynolds in Washington, DC).

PC Tech Journal was unable to obtain the specification. A Compaq spokesperson assured us that the consortium will release it to the "world at large" as soon as it is finished. Unfortunately, because the consortium announced the specification before it was complete, we are all at a disadvantage in a debate that, by definition, must be based on the technical merits.

The supplied technical synopsis highlights the features of an EISA bus. First, existing 8- or 16-bit add-in boards based on industry-standard architecture (a.k.a. classic bus), will fit in an EISA slot. A third connector (the original AT 16-bit extension is the second connector), which is about the size of the original PC's 8-bit connector, is positioned parallel and adjacent to the 16-bit industry-standard connector (this is still under discussion, so the extension connector might be in line by the time the specification is complete). This layout might make a slot wider, although it is impossible to tell without the complete specification. In any event, the ability to use existing industry-standard boards in the new slot is EISA's most important feature and the one underlying its entire design.

Outside of that, the other EISA features more or less duplicate the capability already found in the Micro Channel (see "An Architecture Redefined," David Methvin, August 1987, p. 58). EISA claims to go beyond the Micro Channel in the areas of performance, memory addressing, and software-driven configuration.

On the subject of performance, EISA will provide a 33MB-per-second transfer rate in both the bus-master burst mode and direct memory access (DMA) mode. EISA supporters contrast this rate to the Micro Channel's 20MB bus-master burst mode and 5MB DMA mode. Furthermore, EISA will also allow DMA to access the full 4GB address space (32-bit addresses) compared with the Micro Channel's 16MB (24-bit addresses); note that this is a particular limitation of the Model 80's DMA controller, because the 32-bit Micro Channel bus does support 32 address lines.

Finally, like IBM, which gives board manufacturers adapter IDs that allow the setup software to configure each add-in Micro Channel board properly, the EISA community says it will deliver sophisticated configuration software that uses adapter IDs for EISA boards. The EISA software will go a

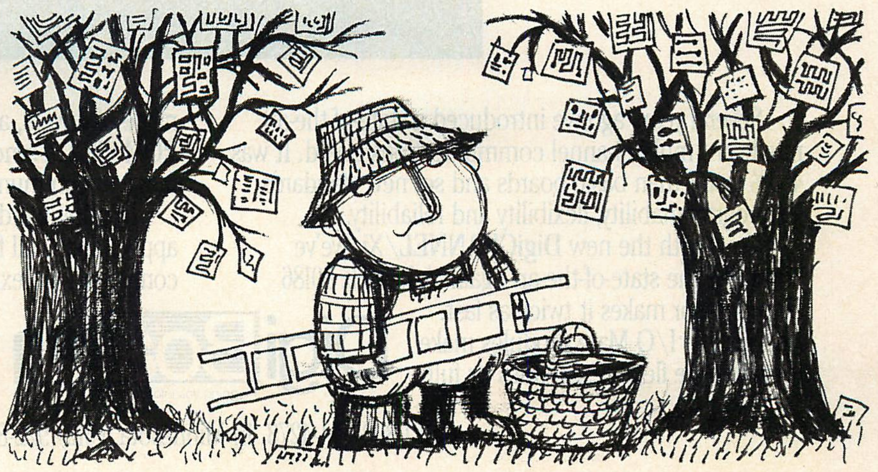


ILLUSTRATION • MACIEK ALBRECHT

step further, however, in managing industry-standard boards that do not have built-in configuration support.

EISA claims several other advantages. An EISA board can be as big as 63 square inches (the standard AT-size board) compared to the Micro Channel's 36 square inches. The EISA specification also calls for delivering more power to each slot, again contrasted with what EISA documents call the "low" power of the Micro Channel slot.

That is about all anybody besides EISA developers knows at this point. *PC Tech Journal* thus concludes that EISA offers only one true advantage over the Micro Channel: its ability to accept existing 8- and 16-bit boards.

ONE PERSPECTIVE

So far, all the EISA announcement has done is spur a major controversy. EISA proponents are hurling claims about a concept for which no implementation will be available for at least 9 months, more likely 12 to 15. Micro Channel proponents, mostly IBM of course, are firing rapid counterclaims.

For the moment, the trick is to understand the claims and counterclaims not in the context that vendors are providing today, but in the more realistic context of the situation 12 months from now. The comparisons the EISA community has drawn between today's Micro Channel and tomorrow's EISA are a bit unbalanced; they should be based instead on what the Micro Channel might become. Predicting where the Micro Channel might go is not terribly difficult because it has much room to grow within the current definition IBM has provided; speculating about its extensions is no less meaningful than discussing the nonexistent EISA bus.

What follows is an analysis of some of the claims being made by both factions, beginning with those by EISA proponents.

- **Buyers today are choosing machines with the industry-standard architecture over Micro Channel machines.**

True, according to Infocorp figures furnished with the EISA press kit, industry-standard machines carry a 76-percent share of the 386 market to the Micro Channel's 24 percent (the matching numbers for the 286 market are 70 percent and 30 percent). The implication EISA advocates want you to take away from such a pie chart is that the reason for the purchase has to do with the bus type.

We disagree. Buyers choose machines primarily on price/performance criteria. Although IBM is improving its line, it is simply not as impressive as Compaq's, especially at the high end, so Compaq sells very well.

At first, buying Micro Channel machines was an act of faith because no third-party boards were available, a critical miscalculation on the part of IBM. In a recent interview with *PC Tech Journal*, IBM Entry Systems President William Lowe acknowledged that IBM made a mistake by not bringing board manufacturers in before it announced the PS/2 family. The situation today, however, is far different. More than 500 Micro Channel boards are available, and, Lowe said, IBM has issued more than 1,300 adapter IDs.

- **Future peripherals will require a higher bus bandwidth.**

We cannot argue this point, because it will always be true. However, it is unlikely that the typical desktop will require the high level of performance suggested by EISA advocates. The high-performance machines will go into niche markets, such as LAN servers, data servers, data acquisition machines, and specialized workstations (CAD, for example).

Compaq has stated for several years now that the AT bus is more than enough for most I/O, especially in Compaq's own Flex architecture, which puts memory on a separate bus, thereby freeing the bus cycles previously used for the CPU's memory access. Virtually any analysis shows that an 8-MHz AT bus in the Flex architecture has plenty of bandwidth left over for higher-performance peripherals. The AT's 4MB-per-second bus bandwidth could allow a 40MB hard disk to be filled in 10 seconds. Does the average user need to fill a disk in 2 seconds with 20MB-per-second bandwidth? No, the typical desktop does not need that level of performance at all.

Unlike EISA, the Micro Channel can accommodate both 16- and 32-bit systems. The architecture's advantages, such as automatic configuration and reliability, accrue to all implementations. In effect, the 16-bit implementation of EISA is the AT bus, which offers none of these advances.

- **Upward compatibility of industry-standard-architecture boards is a customer-mandated feature of EISA.**

There are really two questions lurking here. Does the customer base really want the next bus to be compatible with the industry-standard architecture? Do industry-standard shops use old boards in new machines? (A side issue is that having incompatible buses in a single organization creates support problems such as double spares.)

First, the customer probably does not care if the new bus is compatible with the old. The typical customer is just trying to solve everyday business problems with the best solution. This means finding a cost-effective solution in which all hardware and software components work together, without requiring much support. By the time all is said and done, the bus has vanished under the covers.

Support costs for the Micro Channel are actually lower than for the AT because of the Micro Channel's power-on self test (POST) and automatic configuration capabilities. Dan Lucarini, director of marketing for Tecmar, says that the 25,000 Micro Channel boards his company has shipped represent the lowest customer support costs of any Tecmar product, ever.

Although EISA claims similar automatic configuration capabilities, those will be effective only for 32-bit boards, not for AT-compatible add-ins. EISA manufacturers will supply software that provides configuration emulation for old-style boards, but that is much different from having the physical hardware obey software commands, such as the one that causes a Micro Channel board to shut down while the rest of the system continues to run.

In answer to the second part of EISA's upward compatibility claim, the end-user companies *PC Tech Journal* interviewed said they do not usually propagate older boards into new machines. Although the used PC business is growing, most companies are still in an expansion mode; they are handing down older machines to new users as they upgrade existing users. Although certain types of boards, such as those for fax machines or other special-purpose equipment, might need to stay with the original user, these situations are atypical. In short, choosing the Micro Channel does not really prevent existing boards from being used in the new machine—they will likely remain with the current computer.

- **Micro Channel does not perform as well as EISA.**

Before considering this claim, remember that EISA does not perform at all because it does not yet exist. Theoretically, however, it is true that today's

Micro Channel implementations do not perform to the level projected for the 32-bit, 33MB-per-second EISA. But how does EISA compare with where the Micro Channel is likely to go?

The PS/2 machines of today operate at 20MB-per-second burst mode and 5MB-per-second DMA. A closer look at a Model 80 reveals what might be possible. The 16-MHz Model 80 has a base clock cycle time of 62.5 nanoseconds (ns) for the microprocessor and 125 ns for the DMA controller. The microprocessor can therefore get to memory in the Micro Channel in 125 ns; the Model 80 implementation adds two wait states, for a total of 250 ns. That yields 4 million operations per second, or a bandwidth of 16MB per second on a 4-byte wide (32-bit) bus.

Wait states, however, are not a feature of the Micro Channel design, but rather a feature of the way IBM built the Model 80. Without changing the Micro Channel a bit and staying entirely within its architectural specification, IBM could build a Model 80 with no wait states, yielding a memory cycle time of 125 ns and a bandwidth of 32MB per second. This logic can be extended to bus-master transfers, which can achieve 32MB-per-second rates on a 16-MHz machine and, theoretically, 40MB rates on a 20-MHz machine. This line of thinking also allows DMA transfers to achieve 16MB-per-second rates (or 20MB on a 20-MHz machine), because two bus cycles are required.

To exploit such rates, the components making up a system must be able to operate at those rates. For example, if the memory subsystem cannot keep up, actual performance will fall off and more closely match the maximum rate of the memory. IBM's position is that the cost of a typical desktop computer would be too high if all the components of a system were tuned for the maximum, theoretical rate of the bus.

• **The industry supports EISA.**

The industry certainly seems to be supporting EISA, considering all the vendor endorsements it has received. However, when *PC Tech Journal* called several of these firms to ask what their support for EISA really meant, their short answer was, "Actually, nothing."

From a political point of view, a smaller firm is on dangerous ground if it risks offending bigger players. If EISA succeeds, these firms would lose face, or worse. On the other hand, supporting the concept today, when it is not even necessary to ante up, is totally without risk.

Many of the hardware vendors who say they support EISA are also working on Micro Channel systems. Their reasons are similar to the one Tandy Corporation gave several months ago: "We want to be on whichever bus will take us somewhere."

• **IBM is creating obstacles to building Micro Channel machines.**

This is the patent and royalty issue. IBM's contention is valid: the inventor of a technology should derive benefit from the invention, and those wishing to use it should proffer some form of compensation.

What is peculiar about this rather emotional issue is that several firms already have Micro Channel machines and have therefore reached accord with IBM. Tandy is one. More significant is Advanced Logic Research (ALR), a much smaller company that might be considered to be at a disadvantage in coming to terms with IBM. The fact that small (but growing) ALR is already a player indicates that IBM is not as big an obstacle as IBM's competitors would have us believe. (ALR has also endorsed EISA.)

An important point needs to be cleared up about fees to IBM. The EISA consortium speaks about the specification as if it will be an industry standard apart from IBM. This is not strictly true. IBM recently was awarded some long-pending patents relating to the AT bus (Micro Channel patents are still pending); because EISA incorporates the "industry standard," those building EISA machines will need to reach accord with IBM just as if they were building Micro Channel machines. In fact, Bill Lowe mentioned during our interview that Micro Channel agreements with some vendors are stalled over the issue of retroactive royalty fees for cloned PCs.

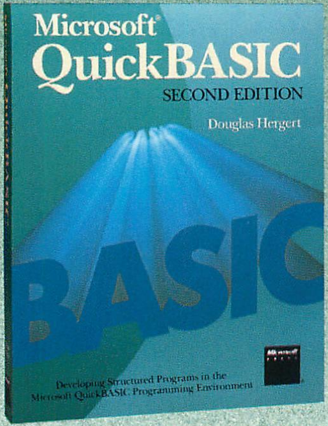
• **The Micro Channel is not an open specification, and IBM could make it less open in the future.**

What the EISA advocates are really saying here is that they cannot build the Micro Channel for free. Concerning openness, the IBM technical documents, while stopping short of giving schematics, do provide all the information a manufacturer would need to build an add-in board, including the vital timing information. That is far more information than IBM ever provided for the AT bus, which the aftermarket readily cloned.

Given IBM's past conservative behavior and legendary marketing tactics, many vendors have the understandable

Continued on p. 35

THE QUICK ROUTE TO QUICKBASIC MASTERY.

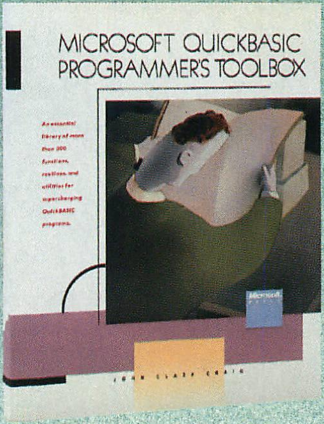


MICROSOFT® QUICKBASIC, 2nd ed.

Douglas Hergert

This new edition of MICROSOFT QUICKBASIC—completely updated for version 4—is a great introduction to all the development tools, features, and user-interface enhancements in Microsoft QuickBASIC. And there's more: six specially designed, full-length programs—including a database manager, an information-gathering and data-analysis program, and a program to create nifty charts—that highlight exciting language elements and reenforce solid structured programming techniques.

$19.95 [Order Code 86-96387]



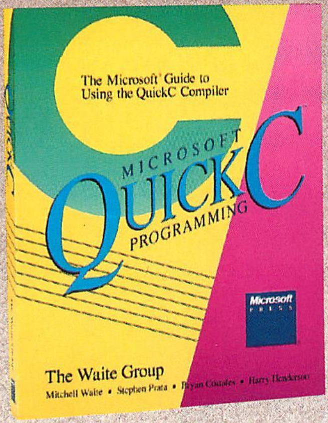
THE MICROSOFT® QUICKBASIC PROGRAMMER'S TOOLBOX

John Clark Craig

An essential library of subprograms, functions, and utilities to supercharge your QuickBASIC programs. The more than 250 subprograms and functions address: ANSI.SYS screen control ■ mouse support ■ pop-up windows ■ graphics ■ string manipulations ■ bit manipulation ■ editing routines ■ engineering, mathematical, and statistical functions ■ game programming ■ interlanguage calling ■ and more. You're guaranteed to turn to this superb collection again and again.

$22.95 [Order Code 86-96403]

SPEAK C MORE FLUENTLY.

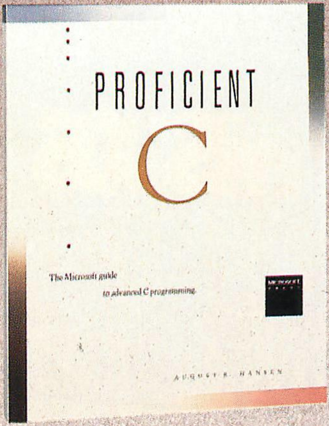


MICROSOFT® QUICKC PROGRAMMING

The Waite Group

Your springboard to the core of the Microsoft QuickC compiler—loaded with practical information and advice on every element of QuickC, along with hundreds of specially constructed listings. Included are the tools to help you master QuickC's built-in libraries; manage file input and output; work with strings, arrays, pointers, structures, and unions; use the graphics modes and the built-in graphics library; control the keyboard and cursor; develop and link large C programs; and debug your source code.

$19.95 [Order 86-96114]



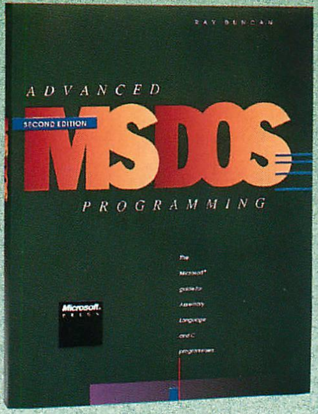
PROFICIENT C

Augie Hansen

If you want to combine C with DOS to produce powerful programs that run at astonishing speeds, start with PROFICIENT C. It's filled a rich assortment of reliable, structured programming methods, techniques, and insights to help you design, code, and test your programs. There's clear information on the PC-DOS development environment, standard libraries and interfaces, and file-and screen-oriented programs. Included are programs that update file modification times, test the driver, view and print files, display non-ASCII text, and scores more.

$22.95 [Order Code 86-95710]

FOR AUTHORITATIVE INFORMATION...LOOK TO THE EXPERTS.

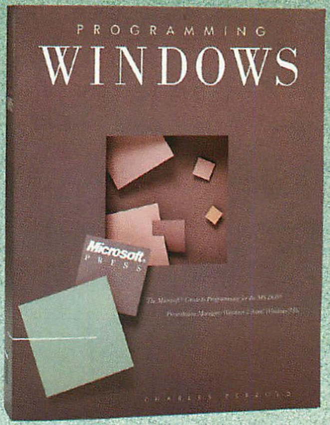


ADVANCED MS-DOS PROGRAMMING, 2nd ed.

Ray Duncan

The preeminent source of MS-DOS information for assembly-language and C programmers—expanded and completely updated. New data covers the ROM BIOS for the IBM PC, PC/AT, PS/2, and related peripherals; MS-DOS through version 4; 4.0 of the LIM EMS, and more. Expert advice plus a healthy assortment of listings covering character devices, mass storage, memory allocation and management, and process management. The reference section, detailing each MS-DOS function and interrupt, is virtually a book within a book.

$24.95 [Order Code 86-96668]



PROGRAMMING WINDOWS

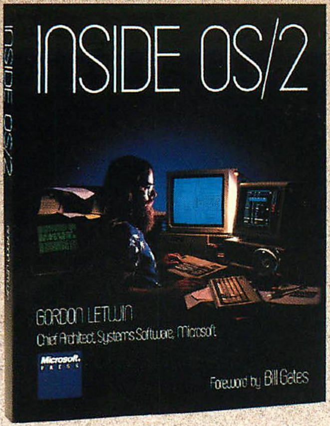
Charles Petzold

Your fastest route to successful application programming with Windows. Full of indispensable reference data, tested programming advice, and page after page of creative sample programs and utilities. Topics include the keyboard, mouse, and timer; icons, cursors, bitmaps, and strings; Windows' memory management; menus; child window controls; keyboard accelerators; using dynamic link libraries; and mastering the Graphics Device Interface (GDI).

$24.95 softcover[Order Code 86-96049]

$34.95 hardcover [Order Code 86-96130]

OS/2...FROM THE INSIDE OUT.

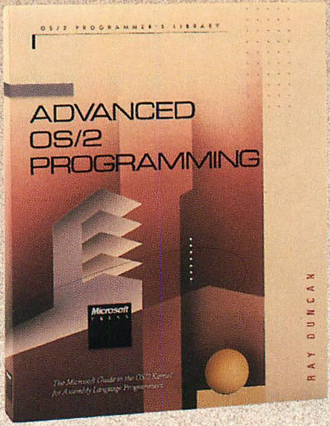


INSIDE OS/2

Gordon Letwin

Here—from Microsoft's Chief Architect of Systems Software—is a candid and exciting technical examination of OS/2. In unprecedented detail, Letwin explores the philosophy, key development issues, programming implications, and role of OS/2 in the office of the future, and provides the first in-depth look at each of OS/2's design elements. You can't get a more inside view of the graphical user interface, multitasking, memory management, protection, encapsulation, interprocess communication, and direct device access.

$19.95 [Order Code 86-96288]



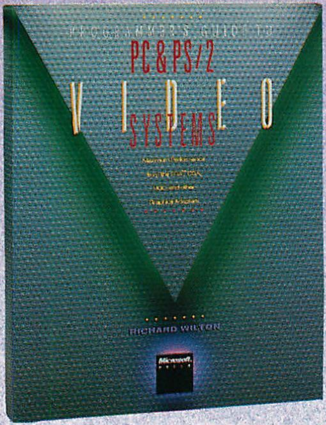
ADVANCED OS/2 PROGRAMMING

Ray Duncan

Authoritative information, expert advice, and great examples in assembly language and C make this comprehensive overview of OS/2 indispensable to anyone serious about programming the OS/2 kernel. Duncan addresses significant OS/2 issues: programming the user interface; mass storage; memory management; multitasking; interprocess communications; customizing filters, device drivers, and monitors; and using OS/2 dynamic link libraries. Plus detailed information on each of the more than 250 system service calls in version 1.1 of the OS/2 kernel.

$24.95 [Order Code 86-96106] Avail. 1/15/89

SPECIALIZED INFORMATION FOR ADVANCED PROGRAMMERS.

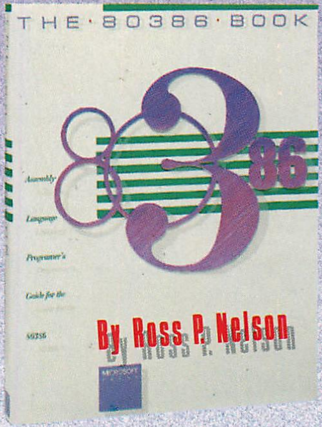


PROGRAMMER'S GUIDE TO PC & PS/2 VIDEO SYSTEMS

Richard Wilton

No matter what your hardware configuration, here is all the information you need to create fast, professional, even stunning video graphics on IBM PCs, compatibles, and PS/2s. No other book offers such specialized programming data, techniques, and advice to help you tackle the challenges of programming to the video hardware. Includes scores of invaluable source code examples. Whatever graphic output you want, do it more effectively with Wilton's book.

$24.95 [Order Code 86-96163]



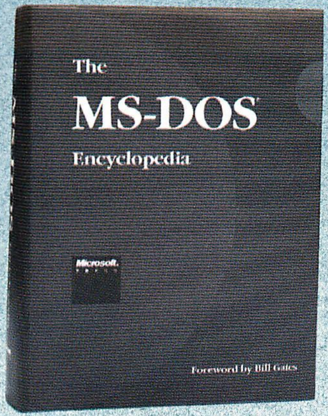
THE 80386 BOOK

Ross P. Nelson

A clear, comprehensive, and authoritative introduction for every serious programmer. Included are scores of superb assembly-language examples along with a detailed analysis of the chip. Topics covered include: the CPU, the memory architecture, the instructions sets of the 80386 microprocessor and the 80387 math coprocessor, the protection scheme, the implementation of a virtual memory system through paging, and compatibility with earlier Intel microprocessors. Of special note is the comprehensive instruction set reference.

$24.95 [Order Code 86-96494]

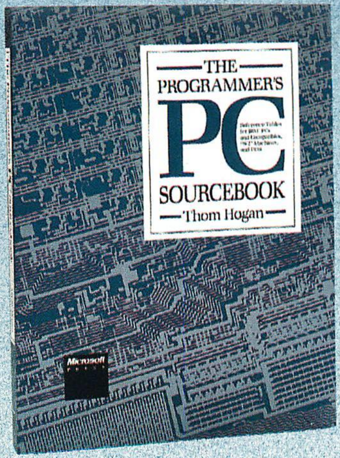
MS-DOS PROGRAMMERS...TAKE NOTE.



THE MS-DOS® ENCYCLOPEDIA

The ultimate reference for insight, data, and advice to make your MS-DOS programs reliable, robust, and efficient. 1600 pages packed with version-specific data. Annotations of more than 100 system function calls, 90 user commands, and a host of key programming utilities. Hundreds of hands-on examples and thousands of lines of code. Plus articles on debugging, installable device drivers, TSRs, Windows, memory management, the future of MS-DOS, and more. Written by a team of MS-DOS experts. Covers MS-DOS through version 3.2, with a section on version 3.3.

$134.95 [Order Code 86-96122]



THE PROGRAMMER'S PC SOURCEBOOK

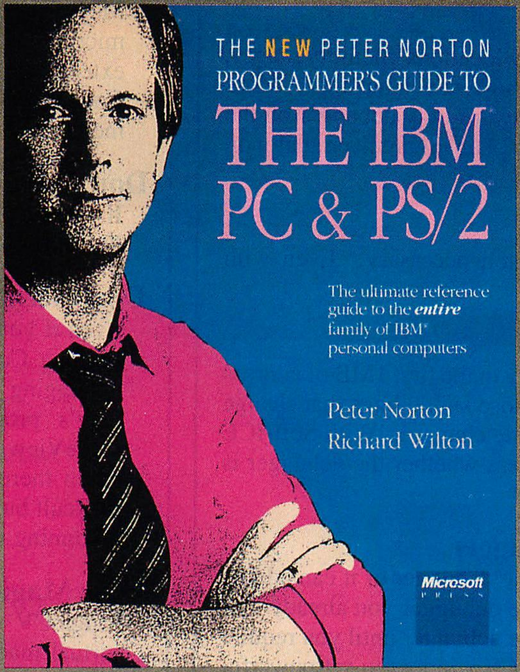
Thom Hogan

At last! Here is factual information — previously published in scores of other sources — organized into one reference. Focusing on IBM PCs and compatibles, PS/2s and MS-DOS, the hundreds of charts and tables cover: ■ numeric conversions and character sets ■ DOS commands, utilities, function calls, and support tables ■ DOS BIOS calls and support tables ■ other interrupts, mouse, and EMS support ■ keyboards, video adapters, and peripherals ■ chips, jumpers, switches, and registers ■ and more.

$24.95 [Order Code 86-96296]

Peter Norton's Best

"... both a text to be read from front to back and a reference book to sit on the shelf waiting with immediate answers to questions ... a must for every serious PC user's technical library." **PC Magazine**



$22.95

THE PETER NORTON PROGRAMMER'S GUIDE TO THE IBM PC has long been a must-have reference for programmers, students, industry experts, even nonprogrammers who want to master the inner workings of IBM microcomputers. Now thoroughly rewritten, THE NEW PETER NORTON PROGRAMMER'S GUIDE TO THE IBM PC AND PS/2 continues to be *the* reference of choice to the new generation of IBM microcomputers—including the PS/2. Accept no substitutes.

Timely New Information

A wealth of updated and new information covers 80286 and 80386 microprocessors; the new PS/2 ROM BIOS; the enhanced keyboard; the new VGA and MCGA; DOS through version 4.0; interrupts, device drivers, and video programming; and programming in C, Microsoft Quick-BASIC, and Turbo Pascal.

Unmatched Programming Advice

Norton and Wilton's characteristically astute advice, programming methods, and techniques will help you create simple, clean programs that are portable among all current—even future—IBM machines. You won't find better authorities—their down-to-earth style makes even the most complex concepts readily understandable.

Solid Technical Data

Included is up-to-date information on ROM software; video, disk, and keyboard basics; clocks, timers, and sound generation; ROM BIOS basics and the ROM BIOS services; DOS basics, interrupts, and functions; and much more. Of special interest are the details of the architectural similarities and differences among IBM machines.

Microsoft® Press

Hardcore Computer Books

*Available wherever books and software are sold. Or call in your credit card order **800-638-3030**. Refer to ad TJB8.*

CIRCLE NO. 145 ON READER SERVICE CARD

SERIOUS DEBUGGING *at a* REASONABLE PRICE

All the speed and power of a hardware-assisted debugger at a software price

Soft-ICE

Hardware-level break points
REAL-TIME break points on memory locations, memory ranges, execution, I/O ports, hardware and software interrupts. More powerful break points than ANY software-only debugger on the market. Soft-ICE gives you the power of an in-circuit emulator on your desk.

Break out of hung programs
With a keystroke - no external switch necessary. Even with interrupts disabled.

Breaks the 640K barrier
Soft-ICE uses ZERO bytes of memory in the first 1MB of address space. This is especially useful for those subtle bugs that change when the starting address of your code changes. With Soft-ICE your code executes at the same address whether the debugger is loaded or not.

Works with your favorite debugger
Soft-ICE can be used as a stand-alone debugger or it can add its powerful break points to the software debugger you already use. You can continue to use your favorite debugger until you require Soft-ICE. Simply pop up the Soft-ICE window to set powerful real-time break points. When a break point is reached, your debugger will be activated.

Solve tough systems problems too
Soft-ICE is ideal for debugging TSRs, interrupt handlers, self booting programs, DOS loadable device drivers, non-DOS operating systems, and debugging within DOS & BIOS. Soft-ICE is also great for firmware development because Soft-ICE's break points work in ROM.

How Soft-ICE Works
Soft-ICE uses the power of the 80386 to surround your program in a virtual machine. This gives you complete control of the DOS environment, while Soft-ICE runs safely in protected mode. Soft-ICE uses 80386 protected mode features, such as paging, I/O privilege level, and break point registers, to provide real-time hardware-level break points.

"Soft-ICE is a product any MS-DOS developer serious enough to own a 386 machine should have."
Dr. Dobb's Journal — May 1988

Both require 80386 AT compatible or IBM PS/2 Model 80. MagicCV requires at least 384K of extended memory. CodeView is a trademark of Microsoft Corporation.

RUN CODEVIEW IN ONLY 8K!

MagicCV

CodeView is a great integrated debugger, but it uses over 200K of conventional memory. MagicCV uses advanced features of the 80386 microprocessor to load CodeView and symbols in extended memory. This allows MagicCV to run CodeView using less than 8K of conventional memory on your 80386 PC.

Don't let 640K be your limit!
If you are closing in on the 640K limit and would like the power of CodeView, MagicCV is for you.

Don't let the debugger hide the bug!
Even if you're not closing in on the 640K limit, running CodeView with MagicCV makes your debugging environment much closer to the end user's program environment. You can use CodeView to locate subtle bugs that only occur when there is plenty of free memory, or those difficult bugs that only occur when your program is running with a couple of TSRs loaded.

How MagicCV works
MagicCV uses the 80386 to create a separate virtual machine for CodeView. MagicCV uses between 4K & 8K of conventional memory as a bridge between the DOS environment and CodeView.

MagicCV is easy to use
If you are a CodeView user, you already know how to use MagicCV too. Just type MCV instead of CV; everything else is automatic.

MagicCV	$199
Soft-ICE	$386

Buy Both and Save $86!

CALL TODAY
(603) 888 - 2386
***or FAX* (603) 888 - 2465**

30 day money-back guarantee
Visa, Master Card and AmEx accepted

NU-MEGA TECHNOLOGIES
P.O. BOX 7607 • NASHUA, NH 03060-7607

Save $86

MagicCV with Soft-ICE
Using Soft-ICE with CodeView gives you the features necessary for professional level systems debugging. MagicCV and Soft-ICE can work in concert with CodeView to provide the most powerful debugging platform you will find anywhere.

concern that IBM might trap them by extending the Micro Channel and taking early advantage of the differences between a new machine from IBM and what it allowed competitors to build. In addition, IBM might not be open with extensions, leaving competitors at a definite disadvantage.

IBM, however, must realize that the vigor in the PC industry comes from the hearty competition its standard has spawned. Even if IBM did decide to play games with the openness of the Micro Channel specification, the aftermarket can always do what it is doing now—attempt to extend the standard in its own direction. After adopting today's Micro Channel, however, the aftermarket would have a better base upon which to build.

• **EISA is what the Micro Channel should have been.**

It is sheer folly to assume that IBM simply walked away from a market of its own creation (which is not to discount the marketing mistakes IBM has made during its transition to the Micro Channel). IBM made the technical and marketing decision that the Micro Channel is a stronger long-term strategy than the AT bus or any extension of it could have been.

We agree. Extending the AT bus means coping with all the problems that are inherent in that bus, a necessity for compatibility. The Micro Channel is a much better design and is easily extensible. As *PC Tech Journal* editor Julie Anderson so often points out, "Standards take you only so far. Then you have to let go to move on."

Now, from the Micro Channel side of the debate:

• **The industry does not need two advanced buses.**

This was IBM's first point in its reply to key customers and dealers after the EISA announcement.

It surely has some self-serving reasons for saying this, but the interests of end-user companies will best be served if they have only one bus to buy and support. Add-in board vendors will not need to dilute their resources to support multiple bus standards. We buyers are the ones who pay for these developments—do we want to pay twice?

• **Micro Channel supports multitasking and networking better than EISA.**

Guess who said that? This is reminiscent of IBM's attempt to link OS/2 and PS/2 together. Curiously, it is quite true: the Micro Channel is better than the AT bus as a hardware platform for a more sophisticated operating system.

The problem with this claim is that there are too many examples of multitasking and networking running on the AT bus today, and running quite well. That makes the claim sound empty.

Furthermore, IBM has done a poor job convincing the market that the Micro Channel architecture is the right solution for running these advanced features. The company needs to find solid ground on which to promote the Micro Channel's system administration features, such as the POST and automatic configuration.

• **EISA is an acknowledgement that the Micro Channel is conceptually sound.**

You bet. EISA contains many of the features of the Micro Channel but no innovations. If the Micro Channel concepts are not good ones, why are they included in EISA?

YOUR POSITION

Fortunately, the buyer has no decision to make today. For all its hype, an EISA machine is about one year away. Before then, *PC Tech Journal* expects that many vendors will enter the market with Micro Channel machines and that IBM will have sold several million more PS/2s. Wider acceptance, the availability of a second source, an active board market, and a rising installed base will all work to reduce resistance to both PS/2 and Micro Channel.

By the time EISA-based machines finally show up, the Micro Channel will have been extended, and EISA may not make much difference at all. It may just be a mistake.

APPLE LOWS AND HIGHS

Apple Computer Inc. continues to make discouragingly slow progress in the corporate marketplace, but it remains instructive to understand the company's new offerings. There is always the chance, after all, of a breakthrough. Recently, Apple introduced machines at both the high end and the low end—and I do mean low.

The Apple IIc Plus indicates how aggressive the company can be when it moves to protect its market share. This revision of the IIc incorporates a 3.5-inch 800KB diskette and switchable clock rates of 1 and 4 MHz. The basic system lists for $675 ($1,099 with color monitor). Those are strong prices for a popular computer that just got much better, and its release is timed just before the Christmas buying season. Very smart. Very aggressive.

Contrast that with the introduction of the Macintosh IIx, which puts the Motorola 68030 processor into the Mac II. A IIx with 80MB hard disk, extended keyboard, and color display and adapter is $11,096.

What is amazing about the IIx is that Apple bills it as posting only a 10 to 15 percent increase in performance over the original Mac II. In today's high-end PC market, performance increments range from 25 to 50 percent, with 25 percent being an absolute minimum; for example, the next version of the 80386 is expected to be a 33-MHz chip, a 25-percent jump over the current 25-MHz ceiling.

More significant than the 68030 is that Motorola built the page memory management unit (PMMU) into the processor, as opposed to making it a separate option for the 68020. That brings the IIx to parity with the 80386.

The most interesting feature of the Mac IIx for those of us on the PC side of the fence is the new 1.44MB diskette drive and Apple's software that allows the Mac to read and write 3.5-inch MS-DOS and OS/2 diskettes in both 720KB and 1.44MB formats; it can also read, write, and format 400KB and 800KB Macintosh diskettes as well as read Apple II ProDOS diskettes. This is the most aggressive sign to date that Apple means serious business.

Apple also introduced a new model of the Mac SE that includes a 40MB hard disk and 2MB of RAM; the system is otherwise unchanged. The $5,069 price is no bargain when compared with AT-class machines. Worse, Apple raised the price of the two other Mac SE models—a move that Wall Street greeted positively, but one that will make it harder to penetrate the office market by making the SE less competitive with its PC counterparts.

SYSTEMS FORUM '89

I have had many calls and letters from readers asking about the next *PC Tech Journal* Systems Forum. Our plans are not yet firm, but I can tell you that the third incarnation of what is becoming an important annual event will be held in the fall of 1989 on the east coast. We largely based our choice of both the time of year and the location on the responses to this year's questionnaire, completed by the attendees at Systems Forum '88 in San Francisco.

Even though the Forum is at least nine months away, planning is under way. You can help. The most critical part of the planning process is the selection of the panel topics that make up the Forum. We would like your ideas. Your suggestions will ensure that the Systems Forum meets your needs and holds your interest. Keep in mind that the Forum will be held next fall, so consider issues that will be timely then, not necessarily today.

Thinking about the tough issues is tough work, so we offer a reward: for any topic we use from your suggestions, we will give the first person suggesting it a 50-percent discount on admission to the Forum or, if unable to attend, a set of session recordings.

Please forward your suggestions to Will Fastie at *PC Tech Journal*, Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044, or via MCI mail or PCTECHline (301/740-8383).

Will Fastie is the editorial director and founding editor of PC Tech Journal. *Executive editor David Methvin contributed to the EISA portion of this column.*

TECH RELEASES

The latest in hardware, software and technology for systems developers and integrators.

SLT/286 laptop from Compaq

SYSTEMS

Equipped with the 16-MHz 80386, the **HP Vectra QS/16** desktop PC is offered by **Hewlett-Packard**. The machine comes standard with 1MB of RAM, either a 1.2MB 5.25-inch or 1.44MB 3.5-inch diskette drive, a diskette/hard-disk drive controller, one serial port, one parallel port, and support for an 80387. The HP Vectra QS/16's 32-bit RAM can be expanded from 1MB to 16MB directly on the processor board. Base model with 1.2MB 5.25-inch diskette drive, $3,995; with a 40MB hard-disk drive, $5,095; with a 40MB hard-disk drive, HP VGA, and choice of 1.2MB 5.25-inch or 1.44MB 3.5-inch diskette drive, $5,495.

HP Vectra QS/16 80386-based desktop computer

Hewlett-Packard Company, Customer Information Center, 19310 Pruneridge Avenue, Cupertino, CA 95014; 800/752-0900

CIRCLE 303 ON READER SERVICE CARD

A full-function portable PC weighing about 14 pounds and based on the 12-MHz 80C286 is available from **Compaq**. The **SLT/286** comes standard with VGA support on a 10-inch-diagonal backlit gray-scale display, a 3.5-inch diskette drive, a hard-disk drive, 640KB of RAM (expandable to 3.64MB), and a removable keyboard with full-size keys. An internal enhanced nicad battery pack provides as much as 3.5 hours of system use. The battery is fully recharged in 1.5 to 3 hours when the system is not in use and can be continuously charged in 6 to 10 hours while the system is connected to the AC adapter. Model 20 (with 20MB hard-disk drive), $5,399; Model 40 (with 40MB hard-disk drive), $5,999. (The SLT/286 was previewed last month in "Compaq's First True Portable," New Directions, Will Fastie, p. 23.)

Compaq Computer Corporation, 20555 FM 149, P.O. Box 692000, Houston, TX 77269-2000; for nearest dealer, 800/231-0900; 713/370-0670

CIRCLE 304 ON READER SERVICE CARD

A small-footprint PC based on the 20-MHz 80386 has been announced by **American Mitac**. Measuring 15-by-16.5 inches, the **Paragon 386E** features two on-board serial ports, one parallel port, a realtime clock, a diskette-drive controller, and 2MB of 32-bit RAM (expandable to 8MB). The Paragon 386E has room for two 5.25-inch diskette or hard-disk drives as well as one 3.5-inch drive; a one-inch-high, 3.5-inch hard-disk drive can be installed internally. The base model has five free slots. The choice of a video controller is optional. Prices start at $4,050.

American Mitac's Paragon 386E

American Mitac Corp., 410 E. Plumeria Drive, San Jose, CA 95134; 800/648-2287; 408/432-1160

CIRCLE 302 ON READER SERVICE CARD

An entry-level 80286-based PS/2 machine from **IBM** is twice as fast as the original Model 30 and offers as much as 25 times more memory capacity. The **PS/2 Model 30 286** features the 10-MHz 286 with one wait state, 512KB of RAM with parity checking (expandable to 4MB on the system board and to 16MB with memory-expansion boards), integrated VGA capability, 1.44MB 3.5-inch diskette drive, and three AT-compatible, full-size slots. Model 30 286-E01 (no hard-disk drive), $1,995; Model 30 286-E21 (20MB hard-disk drive with integrated controller), $2,595. (For a review of the PS/2 Model 30 286, see "The Tenacious 286," David Claiborne, this issue, p. 80.)

IBM Corporation, 900 King Street, Rye Brook, NY 10573; 800/426-2468; 800/447-4700 for nearest dealer

CIRCLE 301 ON READER SERVICE CARD

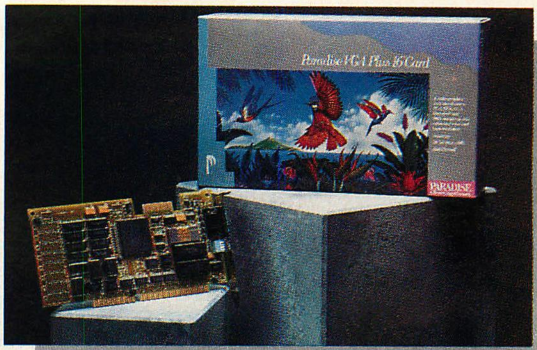
PERIPHERALS

A high-resolution, large-screen, 19-inch analog display is offered by **IBM**. The **PS/2 Monochrome Display 8507** is capable of displaying as many as 1,024-by-768 pixels at 73 pels per inch. It has a viewing area of 14-by-10.5 inches. When attached to the PS/2 Display Adapter 8514/A, it can display advanced text, graphics, and images in as many as 16 shades of gray; 64 shades of gray are possible with the IBM 8514 Memory Expansion Kit. $865.

IBM Corporation, 900 King Street, Rye Brook, NY 10572; 800/426-2468; 800/447-4700 for nearest dealer

CIRCLE 314 ON READER SERVICE CARD

IBM PS/2 Model 30 286



Western Digital's Paradise VGA Plus 16 Card

A scanner recognition board from **Calera Recognition Systems Inc.** for the IBM PC/AT and compatibles can capture text and graphics in a single pass and process as many as 100 characters per second. **TruScan** works with most desktop scanners and facsimile cards and handles typewritten, typeset, laser-printed, dot-matrix, and other

Calera's TruScan scanner recognition system

computer-printed documents. Features include the ability to recognize and format paragraphs, tables, lists, centering, justification, and indenting, as well as bold, italic, and underlined text. It can scan complex multicolumn pages automatically. Model S, $2,495; Model E (with enhanced processing speed and support for rotated pages), $3,495.
Calera Recognition Systems Inc., 2500 Augustine Drive, Santa Clara, CA 95054; 800/544-7051; 408/986-8006
CIRCLE 321 ON READER SERVICE CARD

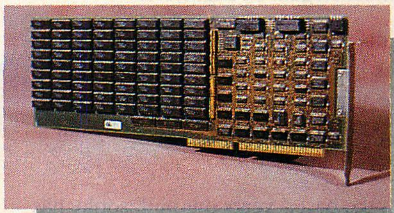
A 16-bit VGA add-in board available from **Western Digital Imaging**, the parent company of Paradise Systems Inc., supports all standard VGA display modes and increased resolutions. The **Paradise VGA Plus 16 Card** provides resolutions as high as 800-by-600 pixels with 16 on-screen colors, or 640-by-400 pixels with 256 on-screen colors. It provides support for 132-column text mode and all pre-EGA standards, including Hercules monochrome graphics. The Paradise VGA Plus 16 Card also incorporates the company's proprietary AutoSense feature,,which detects the peripheral and memory configuration of a PC/AT or 80386-based system to determine whether the product can safely run a 16-bit video BIOS in that configuration. $499.
Western Digital Imaging, 800 E. Middlefield Road, Mountain View, CA 94043; 415/960-3353
CIRCLE 317 ON READER SERVICE CARD

A 4MB memory board for Micro Channel computers has been announced by **Capital Equipment Corporation** (CEC). A resident BIOS automatically configures the **OS/RAM4** board for DOS, OS/2, or Unix. Following installation, no configuration changes are required when changing systems. The OS/RAM4 board provides support for both extended and expanded memory running under DOS. A CEC-designed custom chip serves as a complete Micro Channel interface and memory-support circuit and enables the board to handle expanded memory with no limit on extended memory. 0KB, $395.
Capital Equipment Corporation, 99 S. Bedford Street, Suite 107, Burlington, MA 01803; 800/234-4232; 617/273-1818
CIRCLE 319 ON READER SERVICE CARD

An 80386 system board whose multiple footprints fit the IBM PC/XT, PC/AT, and compatibles is being marketed by **Seattle Telecom & Data Inc.** Available in 16- or 20-MHz versions, the **STD-386 XT** board is configured with a 386, 1MB to 16MB of memory, and eight expansion slots (three 8-bit slots, four 16-bit slots, and one 8-bit slot that can be configured as either an 8-bit expansion slot or 32-bit memory slot). The STD-386 XT board supports an 8-MHz 80287, and it is compatible with operating systems that include OS/2, Unix, Xenix, The Software Link's PC-MOS, and DRI's Concurrent DOS. 16-MHz, $1,750; 20-MHz, $2,550.
Seattle Telecom & Data Inc., 2735 152nd Avenue NE, Redmond, WA 98025-2429; 206/883-8440
CIRCLE 315 ON READER SERVICE CARD

Computer Elektronik Infosys of America Inc. (CEI) has introduced its **RAMFLEX** memory-expansion board, which supports both extended memory and EMS 4.0. The board permits true migration of EMS application software to the OS/2 environment.

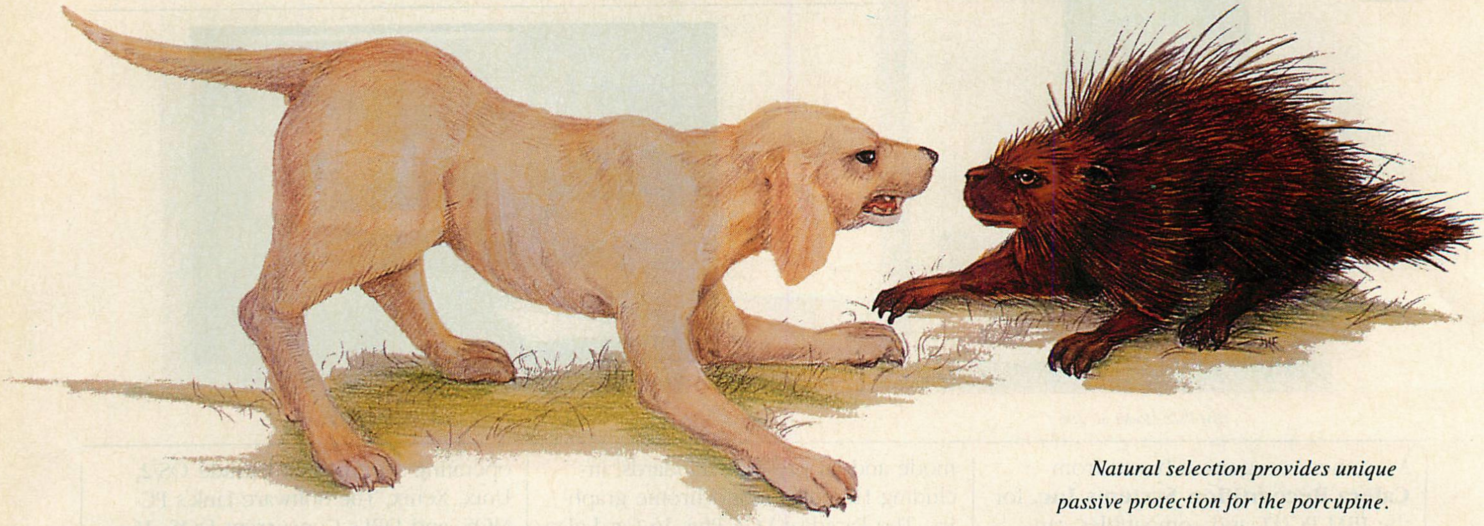
RAMFLEX runs at 12.5 MHz on a standard 16-bit AT bus and uses 100-ns, 1-megabit DRAM DIP memory chips. It provides as much as 8MB of DRAM.



CEI's RAMFLEX memory-expansion board

The board includes several EMS drivers and a look-up table concept that permits EMS programs to run in the DOS compatibility box of OS/2. The board's memory can be configured in 128KB segments into both extended and EMS memory, and it backfills conventional system memory for machines not having 640KB. 0KB, $595; 2MB, $469.
Computer Elektronik Infosys of America Inc., 512-A Herndon Parkway, Herndon, VA 22070; 800/322-3464; 703/435-3800
CIRCLE 318 ON READER SERVICE CARD

The **DiskDoubler** board, now shipping from **Datran**, can double the storage capacity of disk drives without performance loss in the IBM PC, PC/XT,



Natural selection provides unique passive protection for the porcupine.

The Activator - Natural Selection For Software Protection

Inventor and entrepreneur Dick Erett explains how "The Activator" provides sane protection for your intellectual property.

"In any industry, just as in nature, the process of natural selection raises one solution above another. Natural selection is the most elegant of engineers.

In the area of software protection The Block has been selected by the marketplace as the solution that works. Over 500,000 packages are protected by our device.

For the past 4 years our philosophy has been; *'You have the right and obligation to protect your intellectual property.'*

A New Ethic For Software Protection

In allowing end-users unlimited copies of a software package and uninhibited hard disk and LAN operation, The Block has created a new ethic for software protection.

By removing protection from the magnetic media we remove the constraints that have plagued legitimate users.

They simply attach our key to the parallel port and forget it. It is totally transparent, but the software will not run without it.

A New Technology For Software Protection

Our newest model, The Activator, builds on our current patented design, and establishes an unprecedented class of software protection.

We have migrated and enhanced the circuitry of The Block to an ASIC (Application-Specific Integrated Circuit) imbedded in The Activator.

This greatly improves speed and performance, while reducing overall size. Data protection can also be provided.

Programmable Option

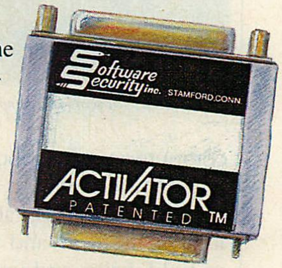
The Activator allows the software **developer** the option to program serial numbers, versions, or other pertinent data known only to the developer, into the circuit, and access it from the program.

Once you program your part of the chip, even we have no way to access your information.

The ASIC makes emulation of the device virtually impossible. It also presents an astronomical number of access combinations.

Full 100% Disclosure

Since The Activator is protected by our patent we fully disclose how it works. Once you understand it, endless methods of protection become evident.



Just as no two snowflakes are the same, no two implementations of The Activator are identical. And like the snowflake the simplicity of The Activator is its greatest beauty.

We never cramp your programming style or ingenuity. Make it as simple or complicated as you desire.

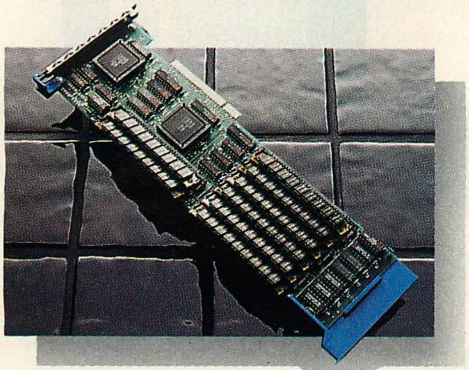
Let us help safeguard what's rightfully yours. Please call today for additional information or a demo unit. *It's only natural to protect your software.*"

1-800-333-0407 ext.102
In Connecticut 203-329-8870
Fax 203-329-7428

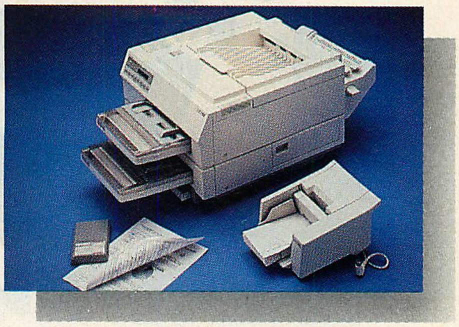
Software Security inc.
870 High Ridge Road
Stamford, CT 06905

CIRCLE NO. 189 ON READER SERVICE CARD

Unlimited Copies • Programmable • No Batteries • Small Size • Fast • Patented • Data Protection



Above Board 2 Plus memory board from Intel



Hewlett-Packard's HP LaserJet IID printer

PC/AT, and other 80286- and 80386-based computers. Using Datran's high-speed data-compression technology, the DiskDoubler automatically and transparently compresses files when they are written to a disk and uncompresses them when they are called from a disk. The board also allows users to back up their hard-disk drive using half as many diskettes as usual. Several utilities are included in the package. $189.
Datran Corporation, 2505 Foothill Blvd., La Crescenta, CA 91214; 800/332-0456; 818/248-8780
CIRCLE 320 ON READER SERVICE CARD

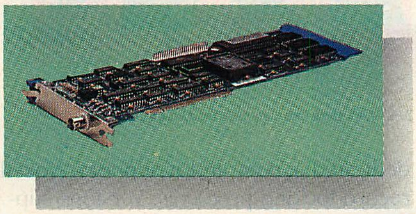
The Personal Computer Enhancement Operation (PCEO) of **Intel** has announced the completely switchless **Above Board 2 Plus**, which provides as much as 8MB of memory—either EMS 4.0 or OS/2—in a single slot for IBM PS/2 Models 50, 50Z, and 60. The Above Board 2 Plus can be populated with a maximum of 2MB, using 256KB single in-line memory modules (SIMMs), or with a maximum of 8MB, using 1MB SIMMs.

The product recognizes memory beyond 2MB by taking advantage of Intel's SoftPROM, a software implementation of the programmable read-only-memory (PROM) technique. SoftPROM ensures that Above Board 2 Plus will be compatible with future releases of OS/2 without costly hardware changes. 0KB, $495; 512KB, $795.
Intel Corporation PCEO, Mail Stop C03-07, 5200 N.E. Elam Young Parkway, Hillsboro, OR 97124-6497; 800/538-3373
CIRCLE 313 ON READER SERVICE CARD

The newest member of **Hewlett-Packard**'s LaserJet family is the **HP LaserJet IID**, which provides a paper capacity of 400 sheets, two paper trays, duplex (two-sided) printing, and an accessory for automatically printing as many as 50 envelopes. Twenty-four fonts come with the HP LaserJet IID printer; 14 fonts are internal to the printer and 10 are included on the HP 92290S2 font cartridge shipped with the printer. The HP LaserJet IID comes with 640KB of memory (expandable to 4.6MB). $4,295; optional envelope feeder, $350.
Hewlett-Packard Company, Customer Information Center, 19310 Pruneridge Avenue, Cupertino, CA 95014; 800/752-0900
CIRCLE 316 ON READER SERVICE CARD

CONNECTIONS

A multiprotocol Micro Channel-compatible communications board has been unveiled by **Network Software Associates Inc.** The **AdaptCoax 2** board enables IBM PS/2 computers to be attached via coaxial cable to an IBM 3274 or 3174 controller. Operating together



AdaptCoax 2 board from Network Software Associates

with AdaptSNA software packages, AdaptCoax 2 supports PS/2-to-host communications through a 3274/3174, using any one of four IBM SNA protocols: LU6.2/APPC, interactive 3270, batch-oriented 3770/RJE, and program-to-program LU0. $595.
Network Software Associates Inc., 22982 Mill Creek, Laguna Hills, CA 92653; 714/768-4013
CIRCLE 311 ON READER SERVICE CARD

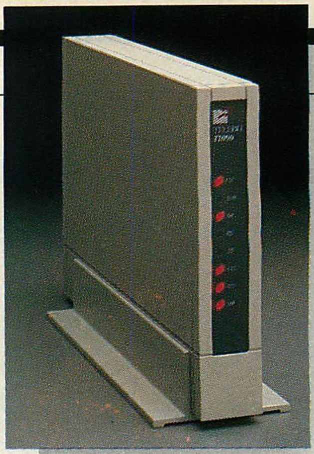
Several LAN analyzer products have been announced by **Network General**. All of the products come with **Sniffer 2.0**, a software package that includes support for a large-packet buffer in EMS 4.0, color coding of protocol layers of the OSI reference model, four sets of filters for station address pairs, a machine language compiler for complex filters and triggers, external triggering and responses, automatic recognition of vendor IDs, and enhanced protocol interpreters.

The **Sniffer Series 500** portable 80386 LAN protocol analyzer includes a Compaq Portable 386 (with a 40MB hard-disk drive and 1.44MB diskette drive), one add-on Sniffer Interface Module, and Sniffer 2.0; when not in use as an analyzer, it can be used as a 386-class PC. The 386 CPU allows the Sniffer Series 500 to complete a seven-layer protocol analysis of a 50,000-frame sample in 30 seconds, and a 1,000-frame sample in less than a second. The Series 500 also features enhanced memory capacity; it captures one-for-one images of a packet stream without packet loss at the highest LAN operating speeds. Sniffer Series 500, including hardware and software, $24,000; additional Series IM-500 Interface Modules, $5,000 each.

The **Sniffer Series MS-500** module- and software-level analyzer includes Sniffer 2.0 and a snap-on Series IM-500 Interface Module, which provides compatibility with one or two selected LAN configurations. Customers supply their own Compaq Portable 386 and perform their own system integration. Each MS-500 Module can be configured for as many as two LAN topologies. One LAN topology, $12,500; each additional LAN configuration, $5,000.

The **Laptop Sniffer Model PA-301** is a laptop protocol analyzer and diagnostic tool for IBM Token-Ring LANs. As with other Series 300 laptop systems (for Ethernet and StarLAN), the PA-301 is housed in a Toshiba T3200 laptop with a 12-MHz 80286 and 40MB hard-disk drive. All Sniffer Series 300

Attachmate's EXTRA! PC-to-mainframe software



Telebit T1000 multispeed modem

analyzers now include 4MB of RAM. Base unit, $15,750.
Network General Corporation, 1945A Charleston Road, Mountain View, CA 94043; 415/965-1800
CIRCLE 308 ON READER SERVICE CARD

Upgrades made by **Attachmate** to its EXTRA! 3270 PC-to-mainframe software support a direct connection to IBM's Token Interface Coupler (TIC) at 4 Mbps—more than 70 times faster than was possible using 56,000-bps SDLC links. Other enhancements to **version 1.3** include improved 132-column display support for EGA, VGA, and 8514 monitors; refined emulation of 3287 host printers to provide faster access to shared LAN printers; light-pen support; and additional memory-management features to free more space for PC applications. $425; upgrade, $75.
Attachmate Corporation, 13231 S.E. 36th Street, Bellevue, WA 98006; 800/426-6283; 206/644-4010
CIRCLE 312 ON READER SERVICE CARD

A multispeed dial-up modem that offers 9,600-bps, error-free throughput over ordinary telephone lines without relying on data compression has been released by **Telebit**. The **Telebit T1000** modem automatically adjusts its operating speeds and modulation schemes to that of the connecting modem; it is compatible with 300-, 1,200-, and 2,400-bps modems as well as high-speed Telebit packetized ensemble protocol (PEP) modems. Error correction is provided by the PEP protocol at speeds of 9,600 bps and by the MNP Class 4 error-correction protocol at 2,400 bps and below. The T1000 includes integrated support for Kermit, Xmodem, Ymodem, and UUCP file-transfer protocols. $795.
Telebit Corporation, 1345 Shorebird Way, Mountain View, CA 94043-1329; 800/835-3248; 415/969-3800
CIRCLE 307 ON READER SERVICE CARD

An industry-standard PC network system that is fully compatible with MS-DOS, the IBM PC LAN program, and other MS-NET-compatible operating systems has been developed by **Datapoint**. The **DATALAN** network program uses the NETBIOS application program interface and works with Ethernet, Token-Ring, ARCnet, and any other LAN hardware for which NETBIOS software is available. DATALAN can run using nondedicated servers or on Datapoint's

DATALAN network system from Datapoint

dedicated server, and it supports Novell record-locking protocols. Versions supporting four, eight, and more than eight users are priced at $595, $1,195, and $1,995, respectively.
Datapoint Corporation, 9725 Datapoint Drive, San Antonio, TX 78284; 800/334-9968; 512/699-7489
CIRCLE 306 ON READER SERVICE CARD

A LAN operating system designed for small work groups of as many as eight concurrent users is offered by **Novell** in its **ELS** (Entry Level Solution) **NetWare Level II 2.12**. The system can run in either 80286 nondedicated mode or 8086 dedicated mode. It provides compatibility for all adapters supported under NetWare 2.1 and includes one remote internetwork connection. The product does not require a key device to operate. NetWare Level II 2.12 includes many features of Novell's SFT and Advanced NetWare operating systems. $1,395.

Also available from Novell is **version 2.12** of its **NetWare** operating system. This version no longer requires a key device, nor is it copy protected. SFT NetWare 2.12, $4,695; Advanced NetWare 2.12, $2,695; upgrades for SFT NetWare 2.1 and 2.11, $235.
Novell Inc., 122 E. 1700 South, Provo, UT 84601; 800/453-1267; 801/379-5900
CIRCLE 305 ON READER SERVICE CARD

A remote, interactive screen-viewing program has been launched by **Quantum Software Systems Ltd.** Running under its QNX realtime multitasking operating system, **Ditto** allows any terminal or computer to access transparently any other computer in a QNX network, providing remote screen viewing of as many as eight virtual screens per computer; it also allows keyboard input into the remote screens. Network and dial-in access requires that only one copy of Ditto be running on the network. Computers running DOS applications under QDOSII, the QNX DOS emulator, can be viewed using Ditto, so that DOS programs can be run from a terminal without modifying the application. $195.
Quantum Software Systems Ltd., Kanata South Business Park, 175 Terrence Matthews Crescent, Kanata, Ontario, Canada K2M 1W8; 613/591-0931
CIRCLE 309 ON READER SERVICE CARD

Enhanced network software from **Digital Research Inc.** allows multiuser environments to run LAN applications. **DR NET 2.0** allows Digital Research's Concurrent DOS 386 2.0 and Concurrent DOS XM 6.0 to run both DOS LAN applications and Concurrent DOS applications across the network. Multiple

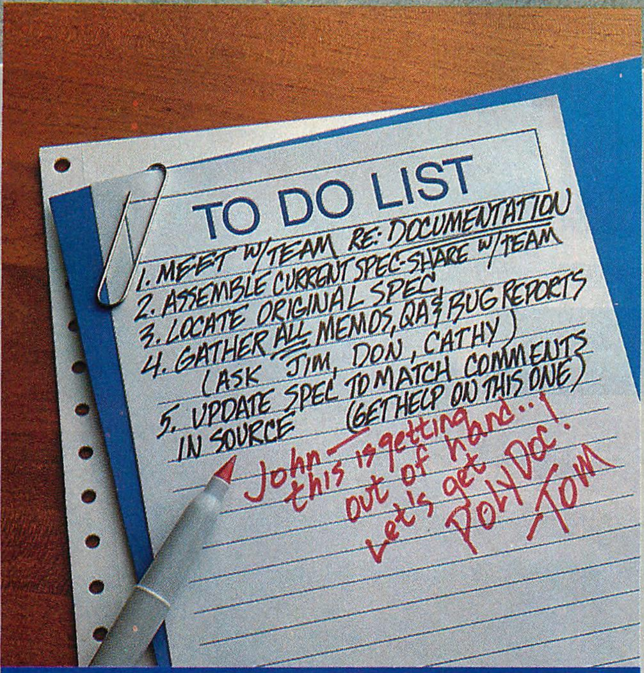
Finally, John Will Get Relief.

"The software is nearly done, what are we going to do about documentation?"

"Let's get John to pull it together from the source comments and specs. It took him forever last time but at least he did it."

John has been selected for the nastiest job in programming — pulling together source code documentation from obsolete specifications, wads of scribbled notes, the ruminations of absent-minded programmers and uncommented code written by a departed team member.

No wonder it's typical for development teams to tolerate incomplete, obsolete or even non-existent documentation. There has been no convenient way to compile, access and share critical code descriptions, change notes, memos and other information. But upon project completion or when a programmer joins or leaves the team, the vital nature of complete and up-to-date documentation is painfully obvious. Now with PolyDoc, programmers, project leaders, teams, and even entire organizations have an easy and practical way to document projects.



PolyDoc.™

The Source Code Documentation Compiler.™

Automatically Generate & Maintain Documentation.

Compiles A Documentation Database

You no longer have to keep separate source code documentation files up-to-date. PolyDoc compiles a documentation database — a PDL (project document library) that stays current with the project as it evolves. The PDL contains source comments, code segments, and explanations automatically extracted from the code and organized according to keywords the programmer has embedded in the code. Bug reports, memos, module and function summaries, QA reports — any supporting document — can be keyed directly to pertinent information at the proper library location. PolyDoc is safe, because source code files are never altered in any way. The library can be viewed on-line or printed at any time. Your documentation becomes a timely, valuable project reference instead of an obsolete, pain in the neck that nobody wants to maintain.

Takes Seconds to Save Hours

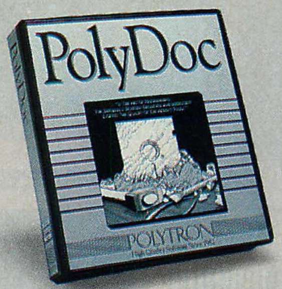
The few seconds required for PolyDoc to scan code and update the PDL will save you hours and even days or weeks compared to manually reconstructing documentation. PolyDoc can scan code to build or update a PDL at the rate of 10,000 lines of code in 60 seconds. Programmers will make the small extra effort to comment code when they see their effort rewarded with a reliable, up-to-date reference for the entire project.

Visually Navigate The Library

Using the interactive screen interface you can view, search, select and optionally print any or all of the PDL. Sections of the PDL are visually cross-indexed and information can be accessed via more than one "path".

Quick Answers to Important Questions

PolyDoc can run "stand-alone" or it can be "memory-resident" so programmers have instant access to the read-only PDL even while using their editor or other software. All team members have instant answers to questions like: "What does this function do?", "Which source file contains this module?", "Who wrote this function?", "What other functions are related to it or do similar things?", "What does the specification document say about this particular feature?", "Is this a reported bug, and who is working on it?".



The Indispensable Workgroup Tool For Programmers and Managers.

Managers appreciate the organization that PolyDoc brings to source documentation. The "chapters" of the PDL can be defined by each project manager, but navigating and searching methods remain constant. The result: a standard method of creating, sharing and referencing documentation on any project.

Use PolyDoc With PolyMake & PVCS

If you are using PolyMake to build your software, you can automatically invoke PolyDoc so any changed modules are scanned. This means your documentation will always reflect the current state of the project. If you are not using PolyMake, you simply make it part of your routine to periodically run PolyDoc over the entire project. To provide a project audit trail and track prior versions of your PDL, the POLYTRON Version Control System (PVCS) can store the differences between PDLs and quickly reconstruct any prior version.

Use On Existing Projects

You can use PolyDoc on projects under development and immediately improve team communication and productivity. You can use PolyDoc on ANY project because PolyDoc can scan and build a PDL for source code written in ANY language.

MS-DOS:

Single User $199; LAN 5-User $647

PolyDoc requires MS-DOS Version 2.0 or later and runs on an IBM PC, XT, AT or compatible. PolyDoc uses 90K of RAM in memory-resident or stand-alone mode. A hard disk is recommended.

30-Day Money Back Guarantee

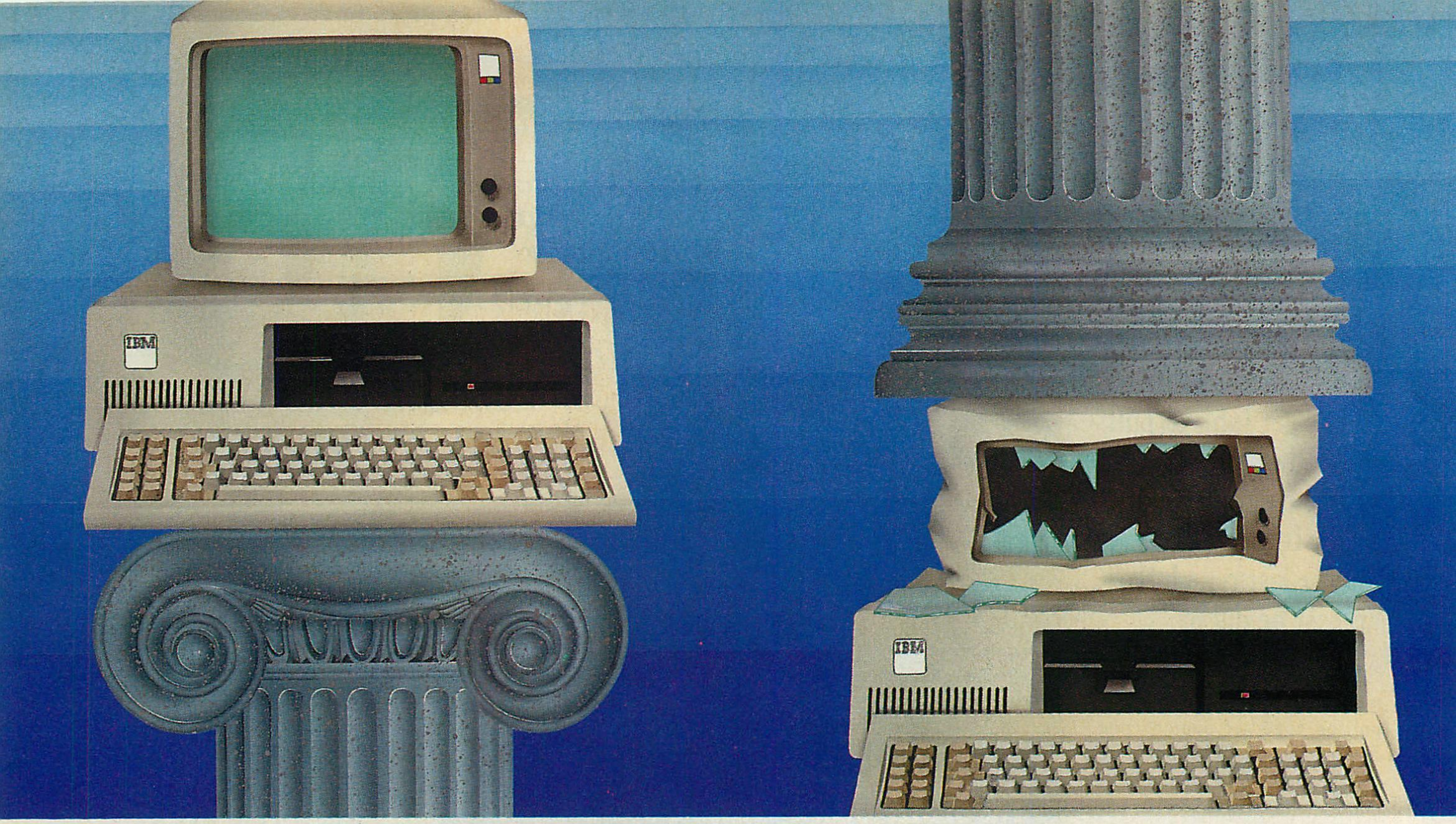
TO ORDER: 1-800-547-4000

Ask for Dept. PTJ. Send Checks and P.O.s To: POLYTRON Corporation, 1700 NW 167th Place, Beaverton, OR 97006. (503) 645-1150. FAX (503) 645-4576 TELEX 325800 POLYTRON

POLYTRON®

High Quality Software Since 1982

CIRCLE NO. 144 ON READER SERVICE CARD



QNX® vs. UNIX OS/2

Architecture can make or break a computer system.

Don't make your systems bear the brunt of massive, monolithic monsters like Unix or OS/2. Instead, build your systems with QNX. The lean, efficient OS that's flexible enough to support any application.

MULTIUSER, MULTITASKING, NETWORKING, AND MORE... QNX is both multiuser *and* multitasking. OS/2 isn't multiuser. Unix may be multiuser and multitasking, but it will hog a huge chunk of your hard disk and system memory. And neither Unix nor OS/2 can do integrated networking.

QNX, on the other hand, hums along using an efficient 150K of RAM, yet provides a powerful multiuser, multitasking, and integrated networking environment.

TRUE DISTRIBUTED PROCESSING. A QNX-based PC LAN lets any user share *any* resource on the network—programs, data, devices, even CPUs—without going through the bottleneck of a central file server. With QNX you get mainframe power at micro prices.

ADDED VALUE FOR VARS. QNX is flexible enough to run on *any mix of* PCs, ATs, PS/2s, with or without dumb terminals. Even diskless workstations are supported. So whether you start small or grow to mainframe proportions, you can easily build and maintain powerful, fault-tolerant systems without spending a fortune on hardware.

REALTIME PERFORMANCE. Only QNX combines the performance of a dedicated realtime executive with the convenience of a rich development environment that includes a host of utilities, C compiler, full-screen editor, symbolic debugger, and multiple full-screen windows.

DOS EMULATION. For those who need their daily dose of DOS, QNX allows a DOS application to run as a single task on each PC on the network.

FREE TECHNICAL SUPPORT. While users of other operating systems have to cough up hefty fees for support, QNX developers enjoy *free* hotline support and *free* BBS access. You'll get prompt answers to your questions and you'll share ideas and insights with fellow developers in the QNX community.

It's time to make the move to QNX.

THE ONLY MULTIUSER, MULTITASKING, NETWORKING, REALTIME OPERATING SYSTEM FOR THE IBM PC, AT, PS/2, AND COMPATIBLES

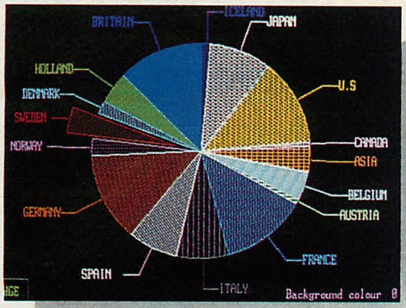
Multiuser	10 (32) serial terminals per PC (AT)
Multitasking	64 (150) tasks per PC (AT)
Networking	2.5 Megabit token passing. 255 PC's and/or AT's per network. Thousands of users per network.
Realtime	4,250 task switches/sec (AT).
Message Passing	Fast intertask communication across the network.
C Compiler	Standard Kernighan and Ritchie.
Flexibility	Single PC, networked PC's, single PC with terminals, networked PC's with terminals. No central servers. Full sharing of disks, devices and CPU's.
PC-DOS	PC-DOS runs as a QNX task.
Cost	From US $450. Runtime pricing available.

QNX

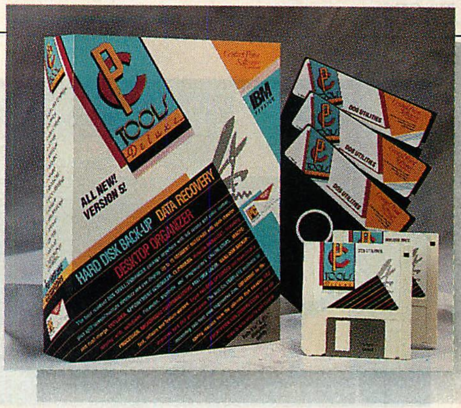
For further information or a free demonstration diskette, please telephone (613) 591-0931.

Quantum Software Systems Ltd. • Kanata South Business Park • 175 Terrence Matthews Crescent • Kanata, Ontario, Canada • K2M 1W8

CIRCLE NO. 181 ON READER SERVICE CARD



Pinnacle Publishing's dGE database graphics screen



PC Tools Deluxe 5.0 from Central Point Software

Concurrent DOS systems can be connected as either servers or requesters, eliminating the need for a dedicated server. Users also benefit from resource sharing of remote disks and printers. Not available separately, DR NET is shipped as part of the Concurrent DOS OEM Redistribution Kit and the Concurrent DOS System Builder's Kit; upgrades will be shipped to current users free of charge.
Digital Research Inc., Box DRI, Monterey, CA 93942; 800/443-4200; 408/649-3896

CIRCLE 310 ON READER SERVICE CARD

SOFTWARE DEVELOPMENT

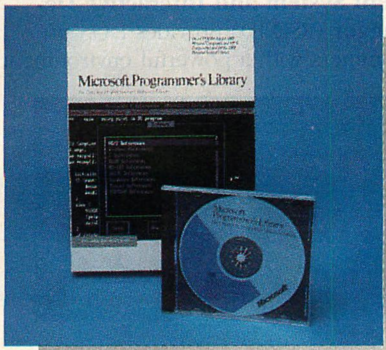
Pinnacle Publishing Inc. has released **dGE**, a database graphics package providing a wide range of high-resolution graphing functions that can be included in the source code of a host program as if they were extensions of the command language. The product comes with interfaces to Ashton-Tate's dBASE III PLUS and dBASE IV, Nantucket's Clipper, Fox Software's FOXBASE+, Wordtech's Quicksilver, dBFast's dBFast, Migent's Eagle, and C compilers. For interpreters and C, dGE has a memory kernel that executes the graphics and a linkage module that is loaded by, or linked to, the host program to provide the interface. The product supports CGA, EGA, VGA, and Hercules monitors. $195.
Pinnacle Publishing Inc., P.O. Box 8099, Federal Way, WA 98003; 800/231-1293; 206/941-2300

CIRCLE 324 ON READER SERVICE CARD

An enhanced user interface and additional programs are included in **PC Tools Deluxe 5.0**, released by **Central Point Software Inc.** The interface features pull-down windows and support for a Microsoft-compatible mouse. The programs include PCSHELL, a superset of the DOS 4.0 shell; PCDESK, a desktop manager; DESchutes, a file encryption and compression program that can encrypt a single file or complete directory with a single command; PCBACKUP, a hard-disk backup program; and COMPRESS, a hard-disk optimizer progam that unfragments files and consolidates a disk's free space. PC Tools Deluxe 5.0 includes a disk cache, an unformat program to recover from an accidentally formatted drive, and on-line help. $79; upgrade, $15.
Central Point Software Inc., 15220 N.W. Greenbrier Parkway, Suite 200, Beaverton, OR 97006; 503/690-8090

CIRCLE 325 ON READER SERVICE CARD

A CD-ROM tool that makes more than 20,000 pages of reference materials and sample code available on-line to programmers is offered by **Microsoft**. The **Microsoft Programmer's Library** database comprises 48 books and technical manuals on Microsoft operating systems and languages. Data are indexed and cross-referenced for fast retrieval and intelligent linking of information. All material can be accessed from within the user's text editor or word processor and copied directly into programs or documents without rekeying. The CD-ROM disk also includes several references for hardware devices, including the Microsoft Mouse, CD-ROM drives, video cards, and a total of 1,200 sample programs. $395.
Microsoft Corporation, 16011 N.E. 36th Way, Box 97017, Redmond, WA 98073-9717; 206/882-8080



Microsoft Programmer's Library CD-ROM reference

CIRCLE 323 ON READER SERVICE CARD

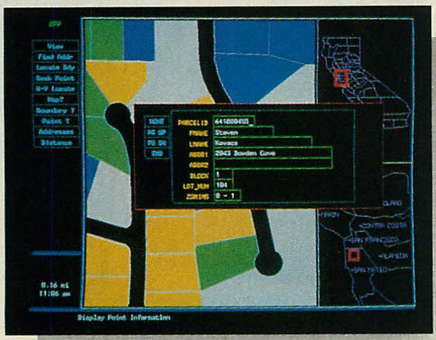
Version 2.0 of **DataPlex**, an intelligent front-end data handler and converter from **Tools & Techniques Inc.**, is available. DataPlex provides a consistent and ergonomic user interface for all data-entry and data-handling tasks. Written in Microsoft C, DataPlex quickly and easily transfers data stored on disk to several formats, including dBASE II, dBASE III, Lotus 1-2-3, all forms of ASCII, and others.

DataPlex dynamically checks all entries against an internal data dictionary and automatically provides edit rules and keystroke macros to speed up and simplify data entry while ensuring data integrity. Because the connections from DataPlex to these formats have been engineered as read/write two-way data paths, the product serves as a central conversion hub for transferring data among applications. Inbound and outbound filter logic is provided to sort, select, edit, and reformat data records while in transit.

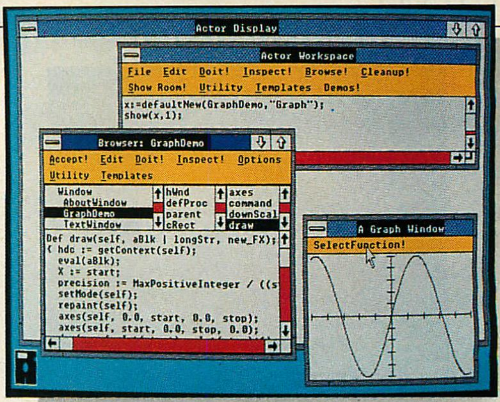
Version 2.0 adds to the formats DataPlex can read from and write to, including DIF, binary/EBCDIC, SuperCalc 4, Enable, Paradox, and R:BASE. The maximum field width has been increased from 46 to 76 characters. $195.
Tools & Techniques Inc., 1620 W. 12th, Austin, TX 78703; 800/444-1945; 512/482-0824

CIRCLE 327 ON READER SERVICE CARD

An interactive disassembler and patcher is being marketed by **RJ Swantek and Associates** for the IBM PC, PC/XT, PC/AT, and compatibles. **Dis•Doc 2.1** disassembles data from RAM/ROM memory or files as large as 500KB. It recog-



Screen from MapInfo 3.0 mapping software



Actor 1.2 screen from The Whitewater Group

nizes the 80386 to 80387 instruction sets. Disassembling takes place at an average rate of 10,000 lines per minute, with as many as 7 passes using more than 20 algorithms to separate code from data.

Dis·Doc 2.1 maintains multiple data segments and inserts MASM data and stack-segment declarations. It can interactively scroll, search the listing or binary image, reformat the listing, adjust the code/data boundaries on-line, rename subroutines and data, and finally create a full or partial listing. Dis·Doc 2.1 automatically comments DOS and BIOS calls, I/O ports, and provides special comments for .EXE structures, driver entry points, and interrupt entry points for the BIOS and first 70 interrupts. A built-in patcher avoids reassembling and relinking large program listings. $129.95.
RJ Swantek & Associates, 178 Brookside Road, Newington, CT 06111-1310; 800/446-4656; 203/560-0236
CIRCLE 328 ON READER SERVICE CARD

Two products available from **MapInfo** are offered to the desktop mapping market. The **MapCode** programming language allows users and developers to create customized desktop mapping applications. MapCode is sold as an optional module to **MapInfo 3.0** desktop mapping software, which now includes every country plotted on a digitized world map, along with demographic data about each country. Using data from existing databases, including dBASE files, MapInfo can automatically create maps that display points on street or regional maps based on their location, as well as thematic maps that highlight user-defined regions, using color, shading, or borders. MapCode, $395; MapInfo 3.0, $750.
MapInfo Corporation, 200 Broadway, Troy, NY 12180; 800/3278-627; 518/274-8673
CIRCLE 322 ON READER SERVICE CARD

An enhanced version of **Actor**, an object-oriented programming language and environment, is shipping from **The Whitewater Group**. Features offered in **version 1.2** include full support for all Microsoft Windows/286 and Windows/386 calls, messages, and styles and as much as 500KB of memory under EMS 4.0. Actor 1.2 also includes multiple application sessions within the limits of available EMS, additional Windows error checking, horizontal scrolling and support for cursor keys on all edit windows, a larger symbol table, and additional documentation. $495; upgrade from previous versions, $75; free to registered users with the purchase of Level 2 (unlimited) support for $250.
The Whitewater Group, 906 University Place, Technology Innovation Center, Evanston, IL 60201; 312/491-2370
CIRCLE 329 ON READER SERVICE CARD

DATABASE MANAGERS

The **Informix 4GL Rapid Development System and Interactive Debugger** (RSD/ID) provides DOS programmers with a powerful environment

Informix Software's 4GL RSD/ID

for developing and simultaneously debugging SQL-based database applications. Released by **Informix Software Inc.**, the product reduces development time by eliminating the need for a C compiler; 4GL code is quickly compiled into pseudocode (p-code), read into memory, and executed by a p-code runner included in the product. RSD/ID, previously available as separate products for the Unix operating system, enables developers to take advantage of as much as 16MB of extended memory to create larger applications. Introductory price until December 31, 1988: RSD/ID plus Informix-SQL, $395; optional six-month maintenance contract, $250. Price thereafter: RSD/ID, $1,495; Informix-SQL, $795.
Informix Software Inc., 4100 Bohannon Drive, Menlo Park, CA 94025; 415/322-4100
CIRCLE 332 ON READER SERVICE CARD

A software development kit for the Ashton-Tate/Microsoft SQL Server, the OS/2-based database solution for networked users, has been announced jointly by **Ashton-Tate** and **Microsoft**. The **SQL Server Network Developer's Kit** includes full code, documentation, and application programming interface (API) libraries for both SQL Server and Microsoft OS/2 LAN Manager. The API libraries can be used with Microsoft's C compiler to create applications under DOS, Windows, or OS/2 that work with SQL over a network. Ashton-Tate is developing version 1.1 of dBASE IV for a future release that will allow developers to build dBASE applications supporting the SQL Server. Introductory price in effect until December 31, 1988, $1,995.
Ashton-Tate Corporation, 20101 Hamilton Avenue, Torrance, CA 90502-1319; 213/329-8000
CIRCLE 330 ON READER SERVICE CARD
Microsoft Corporation, 16011 N.E. 36th Way, P.O. Box 97017, Redmond, WA 98073-9717; 206/882-8080
CIRCLE 331 ON READER SERVICE CARD

All material that appears in Tech Releases is based on vendor-supplied information. These products have not been reviewed by the PC Tech Journal *editorial staff.*

With most PC-host software, you may find yourself connected to more than you'd like.

You could be locked into coax connections. Locked out of future IBM products. Or be chained to a memory-munching monster.

But not with EXTRA! Connectivity Software.™

EXTRA! connects any way you choose: coax, modem or LAN. It works with the latest gateways and LAN hardware, including IBM's new Token Interface Coupler (TIC). Its incredibly long list of features remains the same no matter how you connect. And it saves memory, too.

To learn more, connect with us toll-free: 1-800-426-6283.

Then throw off your chains once and for all.

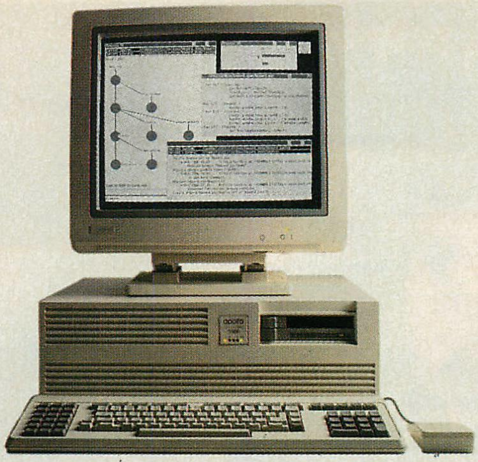
Attachmate

Before Linking PCs To A Mainframe, Think What You Could Be Getting Yourself Into.

CIRCLE NO. 165 ON READER SERVICE CARD

$4990

DN3000 with 15" 1024 x 800 monochrome display



$5990

DN3000 with 19" 1280 x 1024 monochrome display

$8900

DN3000 with 15" 1024 x 800 color display

If you think we don't have UNIX workstation for und

$7990

DN3500 MC68030-based 4-MIPS workstation with 15" 1024 x 800 monochrome display

$8990

DN3500 MC68030-based 4-MIPS workstation with 19" 1280 x 1024 monochrome display

$9990

DN3500 MC68030-based 4-MIPS workstation with 15" 1024 x 800 color display

Open Software FOUNDATION Apollo is proud to be a founding sponsor of the Open Software Foundation.

At Apollo, we believe that the real test of a computer company is not only how often it introduces new technology, but also how quickly it makes that technology available at a truly affordable price.

As testimony to this, we offer not simply one workstation under $10,000, but two entire families of workstations, one of which starts at under $5,000.

Consider the Series 3000.™ A Personal Workstation™ powered like a Porsche® but priced like a PC. $4,990, to be exact. And the new Series 3500.™ An MC68030 machine that brings high performance to a new low. Making it possible for the first time to obtain a

a single high-performance
er $10,000, you're right.

4-MIPS monochrome system for less than $8,000, and a 4-MIPS color system for less than $10,000.

Prices that even include 4MB of memory, a floating-point coprocessor, an IBM PC AT®-compatible bus, and a choice of Ethernet® or Apollo Token Ring networks.

In addition, like all Apollo workstations, these provide support for industry standards such as UNIX® (System V.3 and Berkeley 4.3), PHIGS, X Window System™ and a host of others, as well as network computing and PC compatibility.

The Apollo Series 3000 and Series 3500. Technology as high as the price is low.

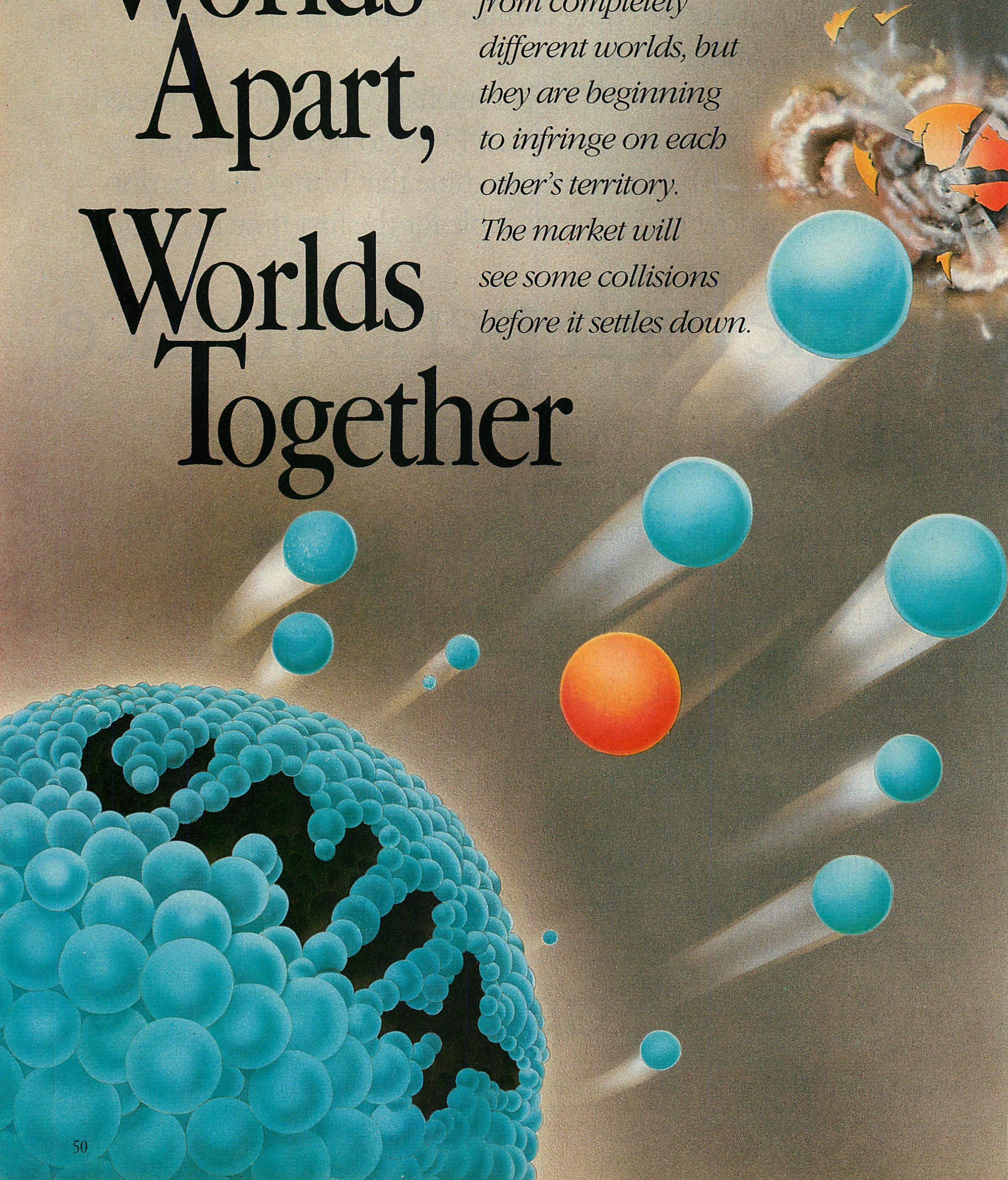
apollo

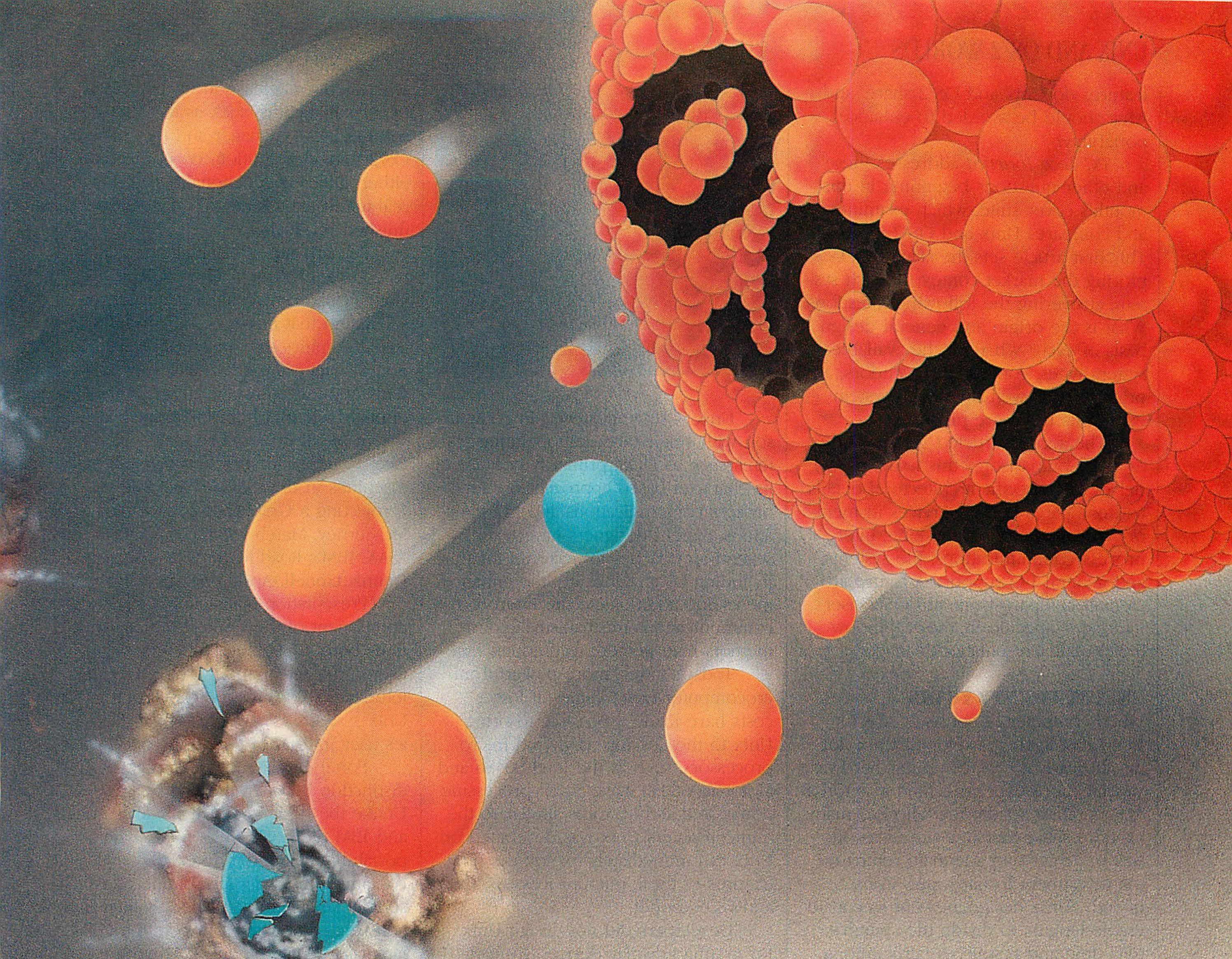
For more information, call 1-800-323-1846, and in Massachusetts (1-800-847-1011) or write Apollo, 330 Billerica Road, Chelmsford, MA 01824.

CIRCLE NO. 109 ON READER SERVICE CARD

Worlds Apart, Worlds Together

Unix and OS/2 come from completely different worlds, but they are beginning to infringe on each other's territory. The market will see some collisions before it settles down.





ROBERT R. MORRIS and
WILLIAM E. BROOKS

The long-awaited arrival of OS/2 has spawned vigorous debate about the future of PC operating systems. Will DOS survive, or will OS/2 signal its demise? To stir the controversy even further, high-performance systems have given Unix a new lease on life since its less-than-spectacular debut into the PC world a few years ago. Developers caught in the middle of this operating system tangle need answers now; sophisticated applications have stretched DOS to its limit.

In the early 1980s, the first desktop computers ran quite efficiently with CP/M and early DOS versions. Hardware evolves much faster than software, however, so despite eight updates in seven years, DOS exploits only a fraction of the capabilities of today's high-performance systems.

Today's high-end PCs potentially have the same computational power that minicomputers had seven years ago. These PCs can now take advantage of operating systems that support large memory spaces, multitasking, and memory protection. OS/2 is the IBM/Microsoft solution to implementing these capabilities on systems based on the Intel 80286 and higher.

Paradoxically, the debut of OS/2 has rekindled or reaffirmed interest in Unix, which has provided many of OS/2's advanced features for almost two decades to an extensive variety of mainframes and minicomputers. Given its portability and proven track record, Unix is attracting developers searching for additional power.

Because Unix and OS/2 provide similar capabilities, developers must weigh one against the other when developing advanced PC-based applications. What makes one system more suitable for a particular application than the other? Should applications be developed for both in parallel? What

ILLUSTRATION • FRANK RILEY

features do Unix and OS/2 share, and what should be avoided to enhance portability?

PC Tech Journal addresses these and other questions beginning in this month's cover suite, which examines and compares Unix and OS/2 in detail. This first article sets the stage by exploring the market context of each system, answering the following questions: What is the origin and general design basis of each? What are the available system types, components, vendors, and pricing? How well does each system exploit its hardware platform?

The second article ("At the Core: An API Comparison," this issue, p. 62) delves into specific architectural details and compares kernel services in memory management, multitasking, interprocess communications, file systems, and general I/O support. A future article will compare Unix and OS/2 user interfaces, specifically the X Window System and Presentation Manager.

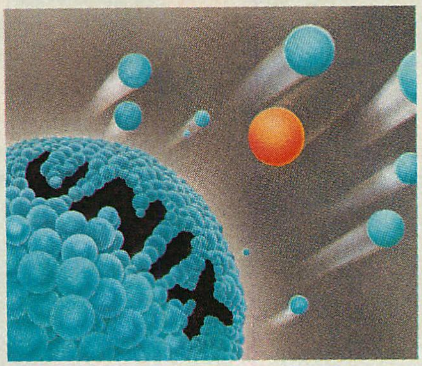
UNIX AT TWENTY YEARS

When computer scientists at AT&T's Bell Laboratories developed Unix during the early 1970s, computer hardware was far more expensive than it is today. An operating system that allowed many users to share machine time was one way to achieve an adequate return on the investment in hardware. Consequently, Unix designers went to considerable lengths to divide the system's CPU time and resources fairly among multiple users.

Developers also emphasized system utilities that lightened the burdens of computer scientists and software developers. Subsequent implementations have continued this trend, making Unix a traditional favorite among software engineers and developers.

AT&T provided Unix source code to the computer-science programs of many universities. Because of this, an entire generation of computer-science graduates cut their teeth on the operating system. As these graduates entered the job market, they brought Unix with them. Because of its availability and openness, Unix continues to be an important operating system for computer-science research.

A remarkable aspect of Unix is the variety of hardware platforms on which it is installed. The interface between Unix, the hardware, and the devices that perform I/O operations is well defined. Therefore the effort required to port Unix from one platform to another can be quantified and priced with relative accuracy.



Unix is slow, however, in comparison with single-threaded operating systems such as DOS, primarily because of its character-based approach to interprocess communications and I/O operations. System designers have taken several approaches to make up for this, including faster hardware and enhancements and additions to the interprocess communications mechanisms and memory-management techniques.

Today's faster, more powerful microcomputer technologies at lower costs have made it possible to bring Unix to the desktop. High-end microprocessors, such as the Intel 80386 and Motorola 68030, have 32-bit processing power and architectures similar to the minicomputers Unix was developed on. Unix can immediately exploit the new features of these microprocessors; OS/2 and DOS cannot.

Developers did not design Unix for graphics-display devices. Instead, Unix is best at supporting many users through low-cost character-based terminals, and most Unix installations still are oriented heavily this way. The Unix community did, however, pioneer the use of graphics in workstations. Unix is the system of choice for popular graphics systems in the high-end scientific and engineering workstation market, including Sun Microsystems and Apollo.

Despite a tremendous amount of research and development, graphics and windows only now are beginning to appear across the range of Unix environments. A number of vendors supply the libraries, drivers, and hardware needed to produce windows and graphics extensions to Unix. One of the most popular packages is the X Window System, developed at the Massachusetts Institute of Technology.

OS/2 COME LATELY

Although Microsoft began shipping OS/2 software development kits in mid-1987, some of the important pieces are still under development—most notably Presentation Manager, which was scheduled for release in October 1988. Other subsystems, such as the Microsoft OS/2 LAN Manager and the IBM OS/2 Extended Edition, do not yet have either sufficient functionality or enough real-world experience to be considered finished products.

Perhaps no other single element so completely sets OS/2 apart from Unix as does Presentation Manager, the subsystem that manages the windows-based graphics user interface. Presentation Manager is the foundation of Systems Application Architecture (SAA), upon which IBM hopes to bring a consistent set of program and user interfaces to applications that run on all the machines in the IBM product line (see "Projecting a Graphics Interface," Ed McNierney, March 1988, p. 54 and "SAA: IBM's Road Map to the Future," Dennis Linnell, April 1988, p. 86).

Presentation Manager is a standard component of the OS/2 system; it defines both a "look and feel" for the user and a set of application program interfaces (APIs) for the developer. Unix systems as a whole lack a graphics user interface of similar scope.

Portability is another important distinction between the two systems. Unlike the designers of Unix, Microsoft and IBM did not develop OS/2 with portability in mind. The OS/2 kernel contains a much larger volume of assembly language than Unix's, which is constructed almost entirely in the C language. Microsoft and IBM wrote OS/2 to exploit (and overcome) the features of the Intel 80286 processor, such as 64KB segment limits, in the hope of obtaining better performance.

Although OS/2 uses the multitasking and memory-management facilities of the 286, it does not exploit new features of the 386 such as paging and virtual-86 mode. Microsoft is developing a 386-specific version of OS/2 but has not announced a delivery date.

GOING TO MARKET

Differences in the market orientations of Unix and OS/2 are clear. Unix vendors bundle extensive sets of software-development and text-processing facilities, while OS/2 vendors, reflecting the applications-oriented tradition of DOS, do not include either development tools or applications with the operating system. Separate products, often from third-party vendors, fill those needs for OS/2 developers.

Unix use on PC-class platforms has been restricted primarily to limited vertical markets—engineering workstations, software-development platforms,

and a limited number of office applications. Because of the real-mode memory restrictions of the IBM PC/XT and slow speed of 6-MHz PC/AT-compatible machines, only a small subset of the engineering tools and business applications developed on Unix-based mainframes and minicomputers are easily ported to the PC running Unix.

For the same reasons, developers often do not write *new* applications for PCs running Unix. Nonetheless, some vendors use the PC and Unix to fill voids that DOS leaves—for example, multiuser business applications for small businesses with a limited number of clerical workers accessing a single PC from several terminals. With the speed and architectural restrictions swept away by the capabilities of higher-speed AT compatibles and 386-based platforms, however, the number of Unix applications moving to Unix-based PCs has increased significantly.

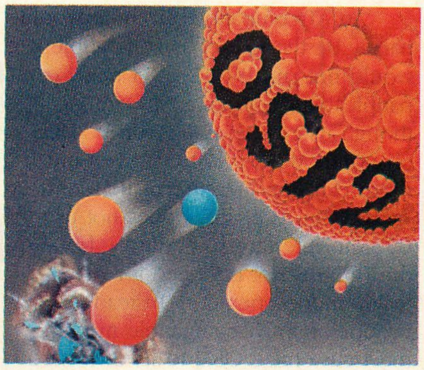
In vivid contrast to Unix, DOS has been successful in all markets. PCs are used not only for general desktop computing but also for special-purpose business, engineering, education, and medical applications. PCs that are packaged to withstand the rigors of industrial environments are even taking over many process-control applications in factory-automation and industrial-control environments—and DOS applications are running on the majority of these machines.

If, as many believe, OS/2 is the natural successor to DOS, it could have the same horizontal market sweep, capturing applications in all areas based on the added functionality that multitasking and integrated graphics can provide. OS/2, however, must overcome many of the same barriers that have plagued Unix on the PC.

Compared with DOS, both OS/2 and Unix are slower and require expensive systems with large amounts of memory. Users would have to upgrade most existing PCs to run either system. The complexity of a multitasking operating system must be hidden from many users to avoid extensive training. Unix systems only recently have addressed the configuration and operation problem; OS/2 is not sufficiently completed or deployed to judge whether it has tamed the problem.

UNIX CHOICES

Comparing two systems as complex as Unix and OS/2 is difficult and is complicated even further because of the wide range of configurations. The current assortment of system variations for



both Unix and OS/2 is diverse—no wonder the developer community so often cries out for standards.

All true Unix variants are based on the original work done by AT&T and Bell Labs. Any company that markets a system with the name Unix, or uses any of AT&T's source code, must license Unix and pay royalties. AT&T, which also markets Unix on its own hardware platforms, retains control over the kernel.

In general, licensees simply can repackage the AT&T Unix code if the platforms are similar enough, adding value as they see fit. The licensee must customize the kernel if it does not support the hardware platform. AT&T allows extensions, and a variety of vendors supply a broad spectrum of enhancements and modifications.

Most versions in use today are based on one of the two main Unix variants—the University of California's Berkeley System Distribution (BSD) or AT&T's System V. Both are descendants of the original system developed at Bell Labs. AT&T has extended System V, its current branch of the Unix family tree, into the realm of commercial applications and capabilities. BSD variants are more inclined to technical, educational, and research-related endeavors.

With System V, AT&T pioneered shared memory, message queues, and semaphore interprocess-communications (IPC) mechanisms. AT&T also established the System V Interface Definition (SVID) and the System V Validation Suite (SVVS), which allow developers to judge if variants are compatible with the System V kernel and services.

Unix systems available for PC-class machines generally are based on the System V branch. Most, however, also have the features and capabilities of the BSD system added as enhancements. System V versions are available for PC-class machines from companies such as Interactive Systems Corporation and Microport Systems Inc. Microsoft, the Santa Cruz Operation Inc. (SCO), and IBM together released another Unix variant, Xenix, that originally was based on AT&T Unix System III, but is now moving toward full compliance with Unix System V.

IBM designed its Unix-based system for the PS/2 product line, called Advanced Interactive Executive (AIX), to be compatible with System V, Release 2. Shipment is set for March 1989. It is derived from the operating system IBM originally offered with its reduced instruction set computer (RISC) RT PC designed for the engineering and scientific market. (See "RT PC: A Significant Departure," Thomas V. Hoffmann, December 1986, p. 56.)

VenturCom offers a version of the Interactive Systems product, modified for better performance in realtime applications. Quantum Software System's QNX is another Unix-like system, not derived from the AT&T product, that is strong in realtime applications.

Sun Microsystems, manufacturer of Unix-based engineering workstations, has been an active participant in developing Unix standards and enhancements. In April 1988, the company announced the Sun 386i, its first 386-based workstation, which runs a Sun version of Unix ported to the 386. Unlike other Unix vendors, Sun sells its version as a bundled hardware and software product.

Several efforts are under way to merge the best features of the Unix variants into a single universal system. The Institute of Electrical and Electronic Engineers' (IEEE) portable operating system environment standard (POSIX) encompasses a set of system calls and library routines but does not restrict variants from supporting other proprietary services.

AT&T and Sun have joined forces to create a system that meets most of the requirements of POSIX and also incorporates many popular features of BSD and Xenix systems. Included in the system is Open Look, a user-interface standard that serves the same purposes as Presentation Manager does for OS/2. Open Look has programming interfaces to X Window and Sun's Network-extendable Window System (NeWS). AT&T and Sun predict that the resulting system, Unix System V Release 4, will be available in mid-1989.

In response, a consortium of leading OEMs, the Open Software Foundation (OSF), was established in May 1988 to support Unix and produce another standardized, open version. Its members include IBM, Digital Equipment Corporation (DEC), Apollo, Hew-

lett-Packard, Toshiba, Stratus, Altos, Groupe Bull of France, and Siemens and Nixdorf of Germany.

Despite the many groups working on Unix, it has fewer variations in the PC world than in the mini- and mainframe world. With the blessing of AT&T, Microsoft produced a Unix/Xenix system that provides full binary (executable file) compatibility with Unix and Xenix and upward compatibility from 286 to 386 versions of Unix.

Presently, Interactive Systems and Microport sell generic Unix System V products ported from the standard AT&T base. SCO provides Xenix System V, a Unix variant that is binary-compatible with both Unix and Xenix. The Interactive Systems and Microport products will be binary-compatible once the companies complete the port of the combined Unix/Xenix system in early 1989. Then all the 386 PC-based versions of Unix will be binary-compatible except IBM's AIX. IBM has not said whether it intends to use the combined Unix/Xenix system in a future release.

Despite a near consensus by vendors on a standard Unix kernel for the PC, products are distinguished by pricing and the availability of bundled utilities. Each vendor provides the base system as a runtime environment and also sells packages for software development and text/document processing, as well as other features for network communications and document preparation/publication. Each vendor's basic runtime system provides the operating-system kernel and most of the standard Unix support utilities and includes the C and Bourne shells (command-line interfaces) as well as installation and system-management utilities.

SCO and Microport offer 286-based versions of Unix as well as 386-based systems; Interactive and IBM sell only 386-based versions. Licensing also differs among products: SCO Xenix and IBM AIX are licensed to support an unlimited number of users; Interactive and Microport sell a base system with licensed support of one to two users and extended licenses to support an unlimited number of users.

OS/2: A DIFFERENT VERSION

OS/2 typically makes its way into the hands of the end user very differently than Unix. Instead of buying the operating system from the company that developed it, an end user obtains the runtime elements of OS/2 through a hardware vendor who licenses it from the developer and either bundles it with the hardware or sells it separately.

An AT-compatible machine would hardly be thought of as compatible if it would not run PC-DOS, yet many compatibles will not run IBM OS/2. Although users commonly run PC-DOS on non-IBM PCs, IBM OS/2 is a much more exacting measure of IBM compatibility than DOS. OS/2 does not use the BIOS to interface to hardware because the BIOS is not designed for protected-mode operation. Instead, each vendor of an AT-compatible machine must customize OS/2 to its own hardware. Third-party vendors of disk drives, tape drives, and other peripherals also must provide OS/2 device drivers with their products if they wish them to operate with OS/2.

To date, no variants of the OS/2 kernel exist—OS/2 has not been around long enough for this to happen.

Unix can immediately exploit the new features of today's high-end microprocessors, whereas OS/2 and DOS cannot.

Moreover, the desire of many vendors to stay compatible with IBM may keep OS/2 variations from ever becoming as plentiful as variants in the Unix world. Most importantly, however, only Microsoft and IBM have access to the source code for the operating-system kernel. Hardware vendors that license OS/2 receive only a Binary Adaptation Kit, which they use to modify the system, primarily by means of device drivers, for specific hardware differences.

One aspect of OS/2's implementation can be misleading—important elements of OS/2 are not part of the kernel. These essentially modular accessories, packaged as dynamic link libraries (DLLs), include Presentation Manager, Microsoft's LAN Manager, and IBM's Extended Edition Communications Manager and Database Manager. OS/2 vendors can bundle their product with subsystems of their own making, in addition to certain standard subsystems.

Most OS/2 subsystem developers have not finished building the first version of their products, while those who have are in only the early phases of introducing the product to the market. It is too early to determine prices and specific details of product bundling.

MEASURING UP

Versions of Unix are available for all Intel 80*x*86 CPUs, as well as many other hardware architectures. OS/2, in contrast, will run on Intel 286 and 386 CPUs only, but does not take advantage of 386 features such as 32-bit registers or demand paging. Both Unix and OS/2 provide preemptive multitasking. OS/2 supports *threads* as a form of multitasking within a process, while Unix supports only processes.

Basic services of Unix and OS/2 are similar; OS/2 even borrows some Unix terminology. File and device-control interfaces compare closely. Both operating systems allow processes to synchronize and communicate with each other using semaphores, pipes, and shared memory.

Both systems run DOS programs. Unix/286 and OS/2 support a single DOS program by switching between the 286 real and protected modes. Unix/386 uses the virtual-86 mode and paging features of the 386 to allow multiple DOS programs to run.

Virtual memory-management facilities that allow large address spaces are supported by both systems. Only Unix takes advantage of the enhanced-memory management facilities of the 386 that provide paging and segments up to 4GB. OS/2 uses the memory-management facilities of the 286, which limit segments to 64KB.

Each system supports shared libraries to reduce the total amount of system memory needed to run multiple tasks that share common library functions. The second article in this cover suite examines the different approaches of Unix and OS/2 to shared libraries.

In keeping with their divergent origins, Unix and OS/2 use different approaches to schedule processes. The Unix algorithm emphasizes equitable sharing of the CPU for better system throughput, while OS/2 stresses quick turnaround to user requests, yielding better response time.

Unix scheduling is based on a complex fairness algorithm normally configured as a round-robin system. Within the kernel, the scheduler adjusts the priority of each task at the end of a one-second system time slice. The system administrator can configure the slice's length. The scheduler places tasks that use the most system resources (such as CPU time) in a low-priority queue. Application programs cannot set specific priorities through system commands or kernel calls, except for a system call that allows a process to lower its own priority.

THE COST OF PROGRESS

What are the costs of moving into the Unix or OS/2 environment? The two primary costs are the direct expense of new software and the time invested to learn the new operating system. Developers who need to maintain DOS versions of their software will spend additional time and money solving portability problems and managing configuration changes.

Costs for the basic kernel of each operating system are similar:

OS/2 Standard Edition	$325
IBM AIX for PS/2	$595
Interactive Systems	$299
Microport 286	$249
386	$299
SCO Xenix 286	$595
386	$695

For each operating system, only the kernel and a minimal set of development tools are provided.

New operating systems require new development tools. At $3,000, Microsoft's OS/2 System Development Kit (SDK) is the most comprehensive set of OS/2 tools available. The SDK documentation, however, is unpolished and contains many errors. An alternative is to purchase the tools separately (see "OS/2 Workshop," Ted Mirecki, August 1988, p. 80). Microsoft bundles its C compiler and Macro Assembler with both the OS/2 and DOS versions; no further expense is required to use these tools under OS/2. Lattice also has taken this approach with its latest C compiler.

Microsoft sells a programmer's toolkit for $350 that provides documentation on the OS/2 API and a set of header files for calling the API from C programs. The operating system, C compiler, assembler, and API documentation sell for $1,275, almost 40 percent of the cost of the SDK.

Unix vendors offer a comprehensive software development kit that includes a C compiler, debugger, source-code management system, librarian, and assembler. Microport's 386 Unix development kit is priced at $549, and Interactive Systems offers its kit for $600. SCO's kit, at $900, includes tools for developing DOS executable files directly under Unix. IBM sells the kit for $175 but does not include the C compiler, which costs $275. IBM also has moved some standard Unix utilities into a separate package, Operating System Extensions, that sells for $250.

These costs are dwarfed by the time needed to learn a new operating system. Unix has an advantage here, because books, seminars, and user groups offer training and support. OS/2 should catch up as it stabilizes and becomes more popular.

—David Methvin

The scheduling algorithm also attempts to resolve system bottlenecks. The scheduler gives higher priority, for example, to a task waiting to complete disk I/O than to one waiting for a buffer. The task about to complete I/O probably will release a buffer or other system resource that can free up one or more blocked tasks.

In contrast, OS/2 cannot allow a fair allocation of system time or resources to multiple users, nor was it designed to do so. OS/2 scheduling is oriented toward the needs of a single-user graphics workstation in which multiple tasks compete for a single set of resources, including the display and input devices that interface the user to the application software.

OS/2 scheduling is based on 32 priority levels within three priority classes: time-critical, regular, and idle (see "Multiple Tasks," Steven Armbrust and Ted Forgeron, November 1987, p. 90). Options in the CONFIG.SYS file permit the end user to change scheduling parameters such as the time slice. Through this approach to scheduling, the designers of OS/2 hope to make the system as responsive as possible, while still giving other tasks the opportunity to run.

Only Unix provides inherent multiuser capabilities. The first user can use the PC's standard keyboard and display, which provides high-speed output. Additional users generally are connected through dumb, character-based terminals by lower-speed asynchronous lines, typically 9,600 bits per second (bps). Sun River sells a product that connects a graphics terminal and mouse to a multiuser Unix PC through a high-speed fiber-optic link, allowing multiple users to run graphics-based applications.

TOOLS OF THE TRADE

The power and cost of tools available for an operating system play a crucial role in determining the feasibility of producing application software (see the sidebar, "The Cost of Progress," above). Even the best kernel has little value if developing applications is too expensive or difficult. Unix and OS/2 shell facilities vary widely. The two most popular Unix shells are the C and Bourne shells. A third, the Korn shell, is emerging as a popular successor but is not as readily available. All three are programmer-oriented interfaces to the kernel that serve as both command interpreters and highly interactive, interpreted programming languages.

The rich features of the various Unix shells (which basically are extremely powerful supersets of the standard DOS shell, COMMAND.COM) are as powerful as many higher-level languages. They include a full set of program-control and conditional-branch instructions and extensive variable manipulation facilities.

Developers incorporate shell commands into shell-script files, which serve as powerful prototyping tools for many different applications, especially in the realm of text processing. Shell commands can access and interact with other programs—data can be piped among shell commands, a wide range of powerful filter programs, and other system utilities. The same user can access multiple activations of each of the shells, running several different shell-script interpretations in the foreground and background. Part of the official repertoire of Unix commands are actually shell-script programs.

Although some Unix shells perform to some extent like Presentation Manager, none has gained wide acceptance. One character-based windowed environment, SCO's MultiView, is bundled with some of the Unix systems available for PC hardware platforms. Its designers, however, made no attempt to provide the coordinated graphics and user-machine interface environment available in Presentation Manager.

The software-development environment for Unix systems includes the C compiler and source-level debugger, as in the standard release from AT&T. Also included are linkers, librarians, and source configuration-control and MAKE facilities. Integrated with the shell facilities are other powerful Unix development tools such as **yacc** and **lex**, which generate programs that perform com-

The Power Of Choice.

When it comes to selecting name brand computers that satisfy a wide range of power requirements, AST makes your job simple.

First, for CAD/CAE, LAN, multi-user and other applications requiring the most powerful personal computer available, AST provides the 25 MHz, AST Premium® 386/25. Based on the Intel® 386 chip with an integrated cache architecture, the AST Premium 386/25 provides the ultimate solution for sophisticated processing.

For those who don't need quite so much power, AST offers the 20 MHz, AST Premium/386C with cache memory and the slim-line, 16 MHz AST Premium Workstation/386SX. Both allow you to use your current MS-DOS® applications. And take advantage of the next-generation, 32-bit operating systems and environments, including Microsoft® Windows/386, UNIX® System V/386, OS/2™ and more.

Of course, not every application requires 386 architecture. For word processing, spreadsheets or desktop publishing, the award-winning AST Premium/286 is the power of choice. Built with FASTslot™ technology and unbeatable compatibility, the AST Premium/286 provides built-in 386 CPU upgradeability when you need it.

And for extra value in a 286 PC that looks like it belongs in the executive suite, choose the AST Premium Workstation/286. As industry critics say, "This is one slick, attractive machine. . . like other AST machines, it's built like a tank."

In fact, AST's reputation for providing high-quality, top-value products extends throughout our leading board-level enhancements and connectivity solutions.

So, give yourself the power of choice—AST Premium Computers. For more information, call **(714) 863-0181**, ask for **operator AA93.**
See full demonstrations at our booth #1588 at Comdex.

☐ Please send me more information on the AST family of Premium Computers.
☐ Please have an AST representative call me.
I am ☐ an end user ☐ a reseller.

Name ____________
Title ____________
Company ____________
Address ____________
City/State/Zip ____________
Phone ( ) ____________

Send to: AST Research, Inc. 2121 Alton Avenue, Irvine, CA 92714-4992.

AST markets products worldwide—in Europe and the Middle East call 44 1 568 4350; in Japan call 813 447 8303; in the Far East call 852 5 717223; in Canada call 416 826-7514.

AST® RESEARCH INC.

Times Have Changed.

plex filtering, lexical and syntactical analysis, and string-parsing functions in applications such as text processing, data reduction, and compiler writing.

The Microport developer's kit, the Software Development System V/386, includes several additional tools, such as the popular Green Hills C compiler. The SCO Xenix Development System has additional DOS cross-development tools and the Microsoft C optimizing compiler. IBM's PS/2 version of AIX provides the standard set of AT&T tools as extra-cost options.

Of the two OS/2 shells, the command interpreter and Presentation Manager, only the former has the programmer orientation of the C and Bourne shells. The command interpreter is much closer in scope and capability to the DOS shell than to a Unix shell. Presentation Manager provides a turnkey approach that relies heavily on windows, graphics, and a mouse. It supports task activation, file selection, and other point-and-shoot functions but relies on the command interpreter to perform batch-file functions.

The OS/2 command interpreter provides more capabilities than does DOS's COMMAND.COM, including significantly more control, conditional branching and command file chaining features. Even with these functions, however, the command interpreter does not begin to approach the breadth of capabilities Unix shells offer.

OS/2's youth is a definite disadvantage in the area of development tools. Vendors have not yet ported their products to OS/2 and, unlike Unix, developers must purchase many of the tools separately. Moreover, OS/2 does not yet have source-maintenance facilities as powerful as those found in Unix's Source Code Control System (SCCS). Optimizing C compilers, source-code debuggers, and MAKE facilities, however, are available now for OS/2 systems (see "OS/2 Workshop," Ted Mirecki, August 1988, p. 80).

Microsoft sells a reasonably complete developer's kit, the OS/2 Software Development Kit (SDK), for $3,000 (additional kits cost $1,500), which includes the operating system itself, C compiler, Macro Assembler, complete user and programmer documentation, and a one-year subscription to the Microsoft on-line software developer's support network.

APPLICATIONS, ANYONE?

The quality and abundance of general-purpose, horizontal DOS applications have undoubtedly spoiled many users, who are in for a shock if they move to either Unix or OS/2 and expect similar surroundings. In the OS/2 world, the problem again is the early stage of development—many vendors have announced OS/2 products but few have delivered. Unix has the bread-and-butter applications (word processing, spreadsheet, data manager) but not the variety available under DOS.

With the Unix Documentors' Workbench, developers can generate elaborate documents by embedding formatting commands into text files with a standard editor and then processing the files with a variety of utilities. Although these text-formatting and document-processing tools are powerful, they are not as interactive as most popular word processors for PCs and are intended for programmers. When integrated with shell-script facilities, however, they are versatile document-preparation tools.

Perhaps no other single element so completely sets OS/2 apart from Unix as does Presentation Manager, the graphics user interface.

Microport, Interactive, and IBM (in AIX for PS/2) provide all of the features of the AT&T Documentors' Workbench as an added-cost option; SCO provides a subset of the package.

As Unix becomes more popular on the PC, familiar DOS applications are beginning to arrive in the Unix world. WordPerfect Corporation's WordPerfect has been available for Unix since the end of March 1988 and Borland International plans to introduce a Unix version of its Sprint word processor by the end of this year. Data managers for Unix are particularly well represented. Vendors or third-party suppliers now sell Unix versions of Relational Technology's INGRES, Oracle Corporation's ORACLE, Fox Software's FOXBASE+, and Informix Software's INFORMIX.

Many vendors have announced OS/2 versions of their data managers, but only a few have been delivered, such as Borland's Paradox and Microrim's R:BASE. IBM is the only vendor that has announced a data-manager facility bundled with OS/2, in this case in the Extended Edition, which will be the basis for OS/2 user access into IBM's Systems Network Architecture (SNA) distributed data-manager facilities.

DOING WINDOWS

All PC-based Unix vendors provide some form of window-manager environment for Unix. Microport supports the MultiScreens package, and Interactive and SCO support MultiView (Microport plans to in the future).

MultiView, a character-based window environment, allows a user to establish multiple windows on PCs and attached ASCII terminals. Each window is a separate tasking environment under the multitasking of Unix and the MultiView window-management system.

The simplest form of windowing, offered by both Unix and OS/2 , is the virtual screen. OS/2 calls a group of processes that share a display and keyboard a *session*; Unix uses the term *virtual terminal* or *virtual console*. Processes in a session or virtual terminal not displayed can continue to run. Under Unix, it is possible to log in different accounts in different virtual terminals and switch freely between them with hot keys. In OS/2, the user switches between sessions with a hot key or from a menu maintained by the Session Manager.

Several vendors have announced implementations of X Window, the burgeoning networked graphics standard. IBM will be offering an implementation for AIX. Interactive Systems plans to build a full implementation that will support EGA and VGA graphics on a PC platform and will operate over a network based on transport control protocol/internet protocol (TCP/IP).

Microport recommends an X Window product from Microfield, a non-networked implementation that uses a high-resolution color graphics board. The Microport Unix system also will support Locus Computing's XSight, which will be an X Window server, supporting EGA and VGA graphics, as well as remote keyboards and mice.

As a graphics standard, X Window does not include all the tools provided in Presentation Manager, such as the dialog box and menu-generation facility. Companies such as DEC and Graphics Software Systems provide tool sets with the missing features.

Microsoft and IBM were expecting to ship Presentation Manager beginning in October 1988. Many owners of the OS/2 SDK already have a prerelease of Presentation Manager that includes a task selector, session manager, and a file-management application. Because

this version contains significant bugs, however, developers are proceeding slowly with applications that use Presentation Manager features.

Two of these features, the API and the graphics-device interface, are significantly more than windows-management systems. These interfaces, and the consistent style prescribed by SAA, effect a single framework to interface end users to applications. Although different vendors will sell these applications for different purposes, the applications can share data, interact with each other, and be displayed and updated concurrently on a single display surface. Keyboard and mouse use is consistent for all applications.

All Unix and Xenix systems support one of the two popular DOS extensions available on 80x86-based Unix systems. Microport supports DOS Merge from Locus Computing (see "The DOS-Unix Union," William Tropp and Stephen Wright, January 1988, p. 78). Interactive Systems and SCO support VP/ix, developed by Interactive and Phoenix Technologies. AIX also will offer DOS windows, which is similar to DOS Merge. Both products support multiple DOS sessions in the 386 environment, in which each DOS application runs in its own virtual 8086 machine. The systems can run concurrent DOS applications in one or more windows and support most applications, including those considered ill behaved.

OS/2 does not exploit the virtual-86 capability of the 386—the user can execute only one DOS session at a time. Because of limitations of the 286 chip, only well-behaved DOS programs run in the OS/2 compatibility box. Programs that attempt direct access to the hard disk, diskette, DMA controller, or interrupt controller often do not run.

OS/2 places its device drivers in low memory, so less memory is available for DOS than in a typical DOS-only system. DOS programs that are not in the foreground do not execute. Communications programs sometimes run erratically in the compatibility box because operating-system overhead can cause them to lose characters.

URGENT COMMUNIQUÉ

Unix generally supports serial communications and terminal-emulation facilities, as well as powerful electronic mail and file-transfer functions between Unix hosts over serial ports. Users access LAN facilities through recent extensions to the Unix IPC facilities. Berkeley originally developed one of the two most popular IPC extensions, the socket mechanism, in BSD release 4.2. This has since been added to many of the System V-based systems.

Sockets, similar to named pipes in OS/2, serve as bidirectional communications paths that can support a variety of protocols. A socket implementation is somewhat restrictive because of complex interactions between the kernel-based socket mechanism and the protocol-handling functions.

AT&T added the *stream* IPC mechanism to System V to make up for several of the limitations of sockets. As in the case of sockets, streams serve as named IPC paths to which various protocol-handling modules are readily attached. Streams are associated with LAN communications paths as well as with simple serial ports and other types of hardware devices.

Compared with DOS, both OS/2 and Unix are slower. They also require expensive systems that use large amounts of memory.

When they are used with terminal drivers, streams have alleviated some of the more restrictive features of terminal I/O. AT&T's modular Transport Layer Interface (TLI) protocol-independent communications ease the interface between higher-level applications-, presentation-, and session-layer functions with the transport- and network-layer functions.

Both BSD and System V LAN protocol implementations normally use the Department of Defense TCP/IP protocols as the network and transport layers of the communications protocol stack. Applications-layer programs are included above those layers to provide mail, file-transfer, remote system-login, and terminal-emulation capabilities.

Several system vendors have standardized protocol implementations above the transport and network layers. Sun's Network File System (NFS) allows transparent access to remote file systems across a network. Sun defined and made public remote-procedure call mechanisms and data-transformation/exchange protocols between disparate nodes, based on the BSD socket mechanism; these serve as the session and presentation layers for NFS.

IBM, Interactive, Microport, and SCO all support network communications with add-on products. (For other Unix companies, such as Sun, networking is standard.) Intelligent LAN controllers handle the requisite protocols, and developers build user applications with drivers and access libraries.

Microport provides a full System V streams-based remote file system (RFS) over intelligent Ethernet boards, which perform functions of the TCP/IP protocol layers. Microport's Unix also can access BSD socket-based extensions through compatibility libraries. Interactive announced a similar product.

SCO's Xenix-NET is compatible with the DOS-based Microsoft Networks (MS-NET). Developers can install intelligent communications boards that support the NETBIOS interface protocol, access various media types and TCP/IP-based networks, and support standard Unix mail- and file-transfer facilities. The Xenix-NET LAN-based distributed file system is not compatible with RFS and NFS virtual file systems.

Microsoft announced that Hewlett-Packard will be porting the OS/2 LAN Manager to Unix, scheduled for early-1989 delivery. The initial release is targeted for the merged Unix-Xenix port developed by Microsoft.

SCO and IBM provide access to SNA-based communications facilities through a 3270 support package. The user establishes multiple, windowed sessions to support micro-to-mainframe communications, file transfer, printer emulation, and extended programmable applications. Third-party add-on packages are available that provide similar capabilities for Microport and Interactive Systems products.

AIX for the PS/2 will include support for TCP/IP networking protocols and X.25 communications facilities that permit a PS/2 machine to become a terminal in a packet-switching data communications network. IBM intends to support the NFS protocols in a future version of AIX.

A lack of standardization in OS/2 communications facilities is exacerbating an already complex communications marketplace. Consequently, some developers are concerned by the rather divergent courses Microsoft and IBM are pursuing to develop communications solutions for OS/2.

Microsoft's OS/2 LAN Manager is evolving along the lines of the current MS-NET facilities. IBM's communications addition to the OS/2 Extended Edition, called the Communications Manager, will provide connectivity be-

tween OS/2-based PS/2 platforms and the SNA communications environment through asynchronous and SDLC-based (327*x*) communications paths. The package supports terminal emulation, file transfer, and some of the functionality of IBM's advanced program-to-program communications (APPC) standards. It also supports asynchronous and synchronous serial adapter boards, as well as a communications coprocessor that will provide multichannel asynchronous and synchronous communications support. IBM plans to offer a version of the Microsoft LAN Manager, called the LAN Server Program, that will interoperate with IBM's DOS-based PC-LAN program.

CAN'T JUDGE A BOOK . . .

All of the Unix/Xenix vendors discussed provide complete documentation for the standard system products and utilities, while documentation and support for OS/2 reflect the unfinished state of that operating system.

Licensed AT&T Unix includes a standard set of documents that describes the shells, utility commands, and programming interfaces. On-line help also is available.

SCO and Microport sell directly to end users and promise to provide full support. Interactive Systems sells 386/ix to OEMs and VARs; end users can purchase it only through distributors and VARs. Interactive has developer-support programs for OEMs and distributors.

SCO holds a significant number of training courses, and Interactive has a complete educational program for all of its products available to OEMs and third-party companies.

OS/2 documentation and support, on the other hand, is in a more confused state. Owners of the Microsoft OS/2 SDK have received more than 20 separate manuals since the prerelease of the OS/2 kernel in the summer of 1987. The first set of manuals delivered (clearly marked as prerelease) contained numerous typographical and technical errors.

Maintaining the OS/2 reference library has become an ugly chore. In the second quarter of this year, SDK owners began to receive the final release of the kernel (version 1.1) and prereleases of Presentation Manager and LAN Manager with a NETBIOS interface. Microsoft replaced some of the original manuals, such as the *User's Guide*, with this shipment; technical references, however, such as the *Programmer's Reference Guide* and the language references, were simply updated. New manuals for Presentation Manager and LAN Manager delivered with the second SDK shipment were in an even more alarming state, including erroneous Presentation Manager coding examples and many other grammatical, spelling, and formatting errors.

The sorry state of these manuals, along with the software bugs, makes it quite clear just how much was still unfinished when Microsoft shipped Presentation Manager to developers. Although Microsoft can fix bugs and documentation errors are always expected in beta releases, policies regarding update and maintenance of large software reference sets can make a significant difference in how easy documents are to use.

Only Unix takes advantage of the enhanced-memory management facilities of the 386 that provide paging and 4GB segments.

TOUGH CHOICES

For developers who need to build applications unfettered by the limitations of DOS, the choices today are anything but clear. Both Unix and OS/2 have strengths that recommend them for some purposes and weaknesses that are handicaps for others.

Until the burden of implementing multiple memory-model applications is lifted from the shoulders of developers, Unix clearly will be superior to OS/2 when run on 386 machines. OS/2's unique (and somewhat restrictive) memory-management techniques limit the ability of OS/2 applications to match the capabilities of similar applications on the Unix/386 platform. Of course, when OS/2 or its descendant is implemented on the 386, many of these limitations will disappear. Performing this move will be significantly more difficult than moving Unix from the 286 to the 386, primarily because of the heavy use of assembly code in the OS/2 kernel and the system's extreme adaptations to the 286 memory-management limitations.

On the other hand, OS/2's Presentation Manager, as it evolves into the full extent of SAA, attacks the windowed graphics application environments in a fashion that Unix cannot match until X Window and other graphics systems stabilize. Graphics-based applications will have reliable human-interface features, providing the end user with a consistency and ease of use now available only with the Apple Macintosh.

Presentation Manager comprises a comprehensive set of features, ranging from windows management to extremely robust graphics programming facilities. Most lacking in Unix's X Window are the powerful, integrated device-independent graphics facilities available in Presentation Manager's graphics programming interface. Although developers can integrate X Window with a number of graphics primitives libraries and high-level toolkits, this is more of a piecemeal approach to a problem that desperately needs a standard solution. Clearly, Presentation Manager represents IBM's desire to provide a standardized workstation interface for the company's wide range of hardware and software products.

Although the worlds in which Unix and OS/2 originated were very different, they are becoming more alike as the systems mature. The older system originally was developed as a lean, no-frills working environment for the computer scientist and professional programmer. It has a wealth of powerful development tools and exploits the power of the 386, but provides little support for end-user applications.

OS/2, on the other hand, was designed specifically for microcomputers where the emphasis is on mass-market applications for nonexpert users. Presentation Manager fits that world well, although OS/2 is still too new to have the stability and broad range of tools that make development easier.

Vendors for both systems, however, are making an effort to provide the missing ingredients. The interfaces necessary for user acceptance are becoming available for Unix, and the most essential tools for developing in OS/2 are now on the market. The best news of all, of course, is that both operating systems unshackle the power of today's high-performance machines. Despite the pros and cons, choosing between Unix and OS/2 is a tough choice—but a happy problem.

Robert R. Morris and William E. Brooks are among the founders of Applied System Technologies Inc. of Ft. Lauderdale, Florida. They have developed graphics user interfaces and distributed, realtime data-acquisition systems for industry.

COVER SUITE:
UNIX vs. OS/2

At the Core: An API Comparison

Not until you dig to the core of Unix and OS/2 can you decide which operating system is best for you. Their API services reveal similar principles implemented in different ways.

ROBERT R. MORRIS and WILLIAM E. BROOKS

Unix and OS/2 may be worlds apart in terms of target audience, marketing strategies, vendor philosophies, and history, but how different and alike are they at their core? Are the application program interfaces (APIs) of both systems similar enough to permit easy porting from one system to another? How can either system be made to look like the other to the hosted application? Could a common interface be developed that would lighten the burden of building system-independent applications? Does one fill a niche better than the other, and vice versa?

These issues are paramount as the two systems now begin to compete for the "beyond-DOS market"—that uncharted territory inhabited by users and developers who are seeking multitasking, larger memory and address space, and potentially more productive applications. (See "Worlds Apart, Worlds Together," this issue, p. 50, for an examination of the respective markets for Unix and OS/2.)

The features and API services provided by Unix and OS/2 are numerous and wide-ranging. API services are those that allow applications to interface with the operating system; the API is actually the package of the many system services that the operating system makes available to a program and the techniques developers use to call them. Comparing selected services—such as memory management, tasking, file I/O, semaphores, pipes, and queues—is an excellent starting point for discerning similarities and differences in the two operating systems.

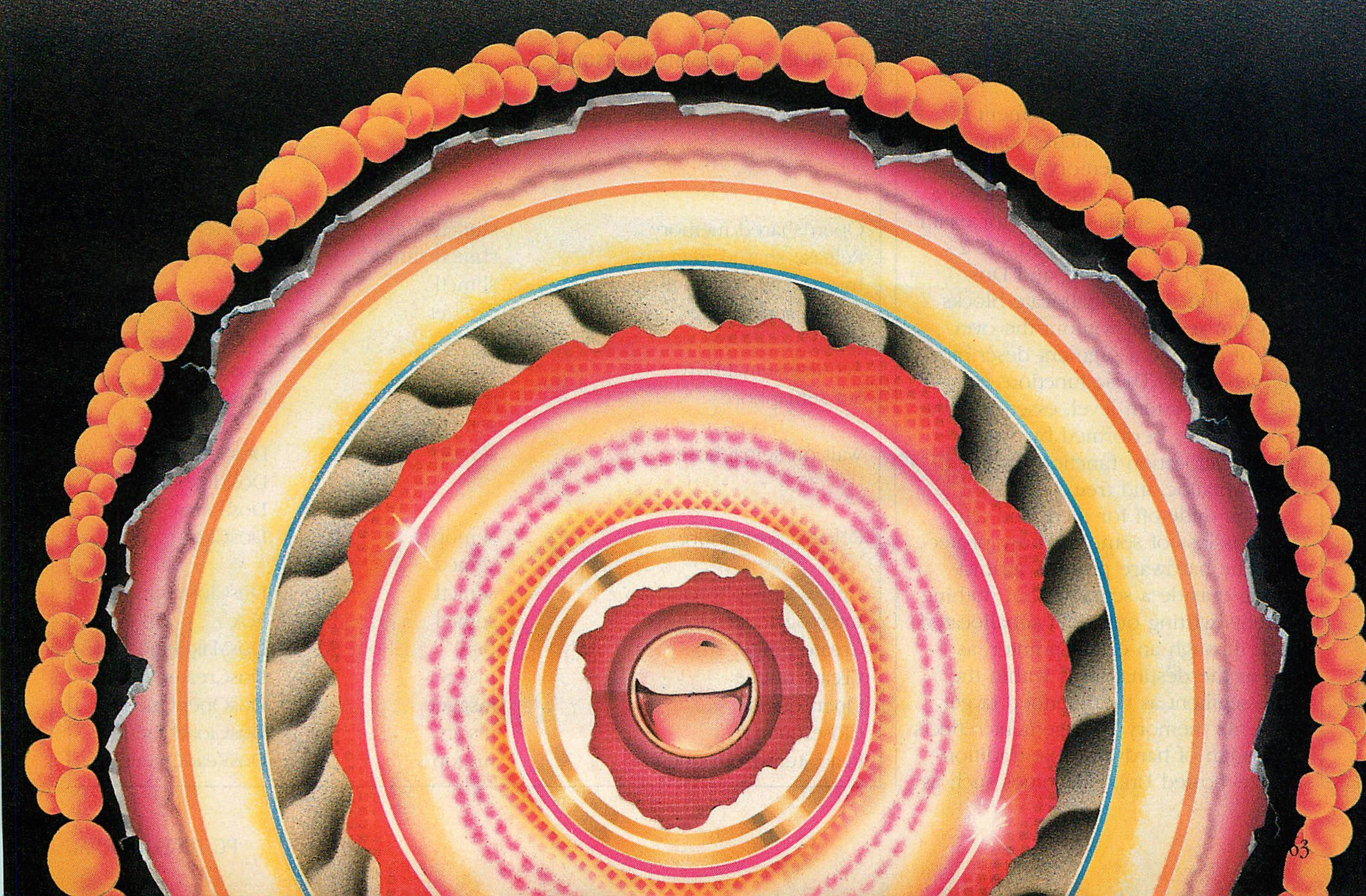
For its comparison, *PC Tech Journal* focuses on selected functional groups of internal services provided by each system: memory management; task control; interprocess communications (IPC), including memory sharing, semaphores, queues, and pipes; file and device-control I/O; and library sharing. Although these selected categories do not represent the universe of operating-system services, they are the ones that are most frequently used and constitute a good basis for comparing and contrasting the APIs of Unix and OS/2. The functions that invoke each service for each operating system are listed in table 1.

PC Tech Journal evaluated AT&T's Unix System V/386, IBM's OS/2 1.0, and Microsoft's OS/2 1.1. Because OS/2 is written for the 80286 microprocessor and Unix System V is designed for the 80386, many Unix advantages are the result of the better chip. An OS/2 version that takes advantage of the 386 chip is not yet available and a delivery date has not been announced. Thus, the comparison of Unix and OS/2 is based on the current versions offered.

CORE OF THE DIFFERENCE

Memory-management facilities in Unix and OS/2 highlight the differences in ideology between the two systems. Many differences are rooted in contrasting philosophies about portability, the simplicity of the interface, and appropriate separation of functions in a layered system architecture. Unix has a

ILLUSTRATION • FRANK RILEY



simpler memory model than OS/2, and it attempts to isolate the developer from hardware differences. OS/2 provides many memory-management facilities but requires the developer to understand underlying quirks and limitations of the hardware.

Because OS/2 is written for the 286, it must deal with memory-management limitations imposed by that chip, one being the 64KB maximum segment size. The DosAllocSeg call allocates a memory segment as large as 64KB, creates a single segment descriptor, and returns its selector. Data objects larger than 64KB are allocated by calling DosAllocHuge, which establishes several descriptors and returns the selector to the first 64KB segment of a multisegment data object. A call to DosGetHugeShift returns a value representing the spacing of the object's descriptors in the local descriptor table.

For example, if DosAllocHuge returns the selector x and DosGetHugeShift returns the value y, the first segment of the allocated data area is addressed by selector x, the second by $x + 2^y$, the third by $x + 2(2^y)$, and so on. The application is responsible for handling or avoiding cases where data records may straddle the boundary between two segments. This task is eased by C and FORTRAN compilers that provide a huge model for automating the use of multisegment data objects, but the code generated is larger and slower than for other memory models (see "FORTRAN Meets OS/2," John Voglewede, this issue, p. 96).

Each allocation of a segment by the operating system incurs some overhead in time and memory, so it becomes impractical to use DosAllocSeg to create many small objects (for example, 10-byte records). Instead, OS/2's DosSubSet, DosSubAlloc, and DosSubFree functions suballocate blocks from one segment without the overhead of creating a segment descriptor for each block. These functions provide, at the system level, essentially the same services performed by the memory-allocation functions of the C library, **malloc** and **free**.

The trade-off for increased efficiency is loss of some memory protection. The hardware protects against writing outside a segment, but nothing prevents writing outside a suballocated block through an errant pointer, as long as the destination is within the same segment as the intended target.

Unix memory management reflects the design of hardware that traditionally has hosted Unix. Machines such as

TABLE 1: ***Selected Unix and OS/2 Services***

	UNIX	OS/2
MEMORY MANAGEMENT		
Allocate memory	brk/sbrk	DosAllocSeg DosReallocSeg
Free memory	brk/sbrk	DosFreeSeg
Huge-memory segments	N/A	DosAllocHuge DosReallocHuge DosHugeShift
Buffer-pool management	N/A	DosSubSet DosSubAlloc DosSubFree
Disposable memory	N/A	DosUnLockSeg DosLockSeg
Memory locking	plock	Link options
Create code segment	sysi86	DosCreateCSAlias
PROCESS AND THREAD CONTROL		
Execute a file	exec fork ptrace wait pause	DosExecPgm DosCWait
Create a new process	fork	DosExecPgm
Create/control thread	N/A	DosCreateThread DosSuspendThread DosResumeThread
Get process ID	getpid	DosGetPID
Set process priority	nice	DosSetPrty
Get process priority	N/A	DosGetPrty
Terminate a process	kill	DosKillProcess
Exit from a process	exit	DosExit DosExitList
Debug a process	ptrace	DosExecPgm DosPTrace
INTERPROCESS COMMUNICATIONS		
Allocate shared memory	shmget	DosAllocShrSeg DosAllocSeg DosAllocHuge DosGiveSeg DosGetSeg
Open shared memory	shmctl/shmat	DosGetShrSeg
Release shared memory	shmctl/shmdt	DosFreeSeg
Interrogate shared memory	shmctl	N/A
Create semaphore	semget	DosCreateSem DosOpenSem
Acquire semaphore	semop	DosSemRequest
Set semaphore	semop semctl	DosSemSet
Wait on semaphore	semop	DosSemWait DosMuxSemWait DosSemSetWait
Release semaphore	semop semctl	DosClear
Delete semaphore	semctl	DosCloseSem
Interrogate semaphore	semctl	N/A
Create pipe	pipe	DosMakePipe
Create queue	msgget	DosCreateQueue
Open queue	msgget	DosOpenQueue
Close queue	msgctl	DosCloseQueue
Read item from queue	msgop	DosReadQueue

	UNIX	OS/2
INTERPROCESS COMMUNICATIONS *(Cont.)*		
Write item to queue	msgop	DosWriteQueue
Read item without removing	N/A	DosPeekQueue
Interrogate queue	msgctl	DosQueryQueue
Order queue by:		
Priority	●	○
LIFO	○	●
FIFO	○	●
Type	●	○
FILE I/O		
Set working directory/drive	chdir	DosChDir DosSelectDisk
Get working directory/drive	getcwd	DosQCurDir DosQCurDisk
Create a directory	mkdir	DosMkDir
Remove a directory	unlink	DosRmDir
Search paths for file	N/A	DosSearchPath
Find/read directory	N/A	DosFindFirst DosFindNext DosFindClose
Create a file	creat	DosOpen
Delete a file	unlink	DosDelete
Open a file	open	DosOpen
Read a file	read	DosRead DosReadAsync
Write a file	write	DosWrite DosWriteAsync
Change file-pointer position	lseek	DosChgFilePtr
File/record locking	fcntl chmod	DosFileLocks
Close file	close	DosClose
Duplicate a file handle	dup	DosDupHandle
Flush buffers	sync	DosBufReset DosOpen option
Change file-access mode	chmod	DosSetFileMode
Change owner of a file	chown	N/A
Get file status	stat fstat access	DosQHandType DosFindFirst DosFindNext DosFindClose DosQFileInfo DosQFileMode
Set access/modification times	utime	DosSetFileInfo
Get file-system status	ustat statfs fstatfs	DosQFSInfo
Get file-system information	sysfs	DosQFSInfo
Change file-handle table size	N/A	DosSetMaxFH
Set file-protection mask	umask	N/A
Mount a file system	mount	N/A
Dismount a file system	umount	N/A
Link to a file	link	N/A
Set write verify	N/A	DosQVerify DosSetVerify

● *= Yes* ○ *= No* *N/A=Not applicable*

Unix and OS/2 provide an immense number of API services. Comparing selected services sheds light on the differences between the two systems. Unix and OS/2 are functionally equivalent, but their approaches are often worlds apart. Many API differences stem from architectural differences in 286 and 386 microprocessors.

the DEC VAX and the Motorola 68000 provide a large unsegmented address space to a process. The 386 is the first in the Intel family that could provide a similar environment. In a Unix process, the code, data, stack, and shared data are mapped into the same segment. Therefore, Unix has no equivalent to the DosAllocHuge function. On a 386-based platform (the first in the Intel family to provide a flat address space), this feature is not needed. Versions of Unix for 286 systems ignore the problem and leave it to developers to solve.

Unix provides two system calls, **brk** and **sbrk**, to allocate and deallocate memory. These services are simple and inflexible: memory must be added at the end of the process's currently allocated space and must be deallocated by releasing memory from the end of that space. Rather than handling the details of memory-block suballocation in the kernel, Unix leaves this task to routines in the standard C library. Other languages under Unix provide similar library routines to allocate memory.

In both Unix and OS/2, programmers can forego using the operating-system services in favor of the C library functions **malloc** and **free**. The memory pool that Unix and OS/2 processes use through the C library is called the *heap*. Using the library functions to manage the heap increases application portability between Unix, OS/2, and DOS. It is rarely necessary or desirable, therefore, to call the system-level functions in either Unix or OS/2.

The difference between the implementation of **malloc** in Unix and OS/2 arises from the limited segment size of the 286. OS/2 C compilers provide a *near heap* in the default data segment for small data objects, and a *far heap* that can accommodate data in other segments. Special library functions, **_nmalloc** and **_fmalloc**, allocate objects in these heaps. The **malloc** function is identical to **_nmalloc** for near-data (small and medium) memory models, and to **_fmalloc** for far-data (compact and large) models.

For better efficiency in memory allocation and accessing data in OS/2, the developer can use *mixed-model* programming. Using the **near** and **far** key words, the programmer can explicitly specify the pointer size for a particular object, overriding the default size. The advantage of mixed-model programming is that the compiler can generate smaller and faster code for near pointers, using less-efficient far pointers only where necessary. The disadvantage is that the developer becomes more

involved in managing memory and must contend with complex combinations of pointer types.

In Unix, the unsegmented address space of the 386 allows a single **malloc** function to allocate objects of any size. When a Unix process requires more memory than is available in the heap, **malloc** calls **sbrk** to expand the data space of the process. When the process calls **free** to deallocate memory, the memory is simply returned to the heap's free space—not to the system.

Because the address space is large, processes are not likely to run out of memory in this virtual system. However, the heap space could become fragmented, with still-allocated data separated by space now available for reallocation by **malloc**. The paging scheme can ensure that only pages with frequently referenced data remain in physical memory.

In terms of memory management, the benefit of Unix over OS/2 to high-level language programmers is the convenience of programming all applications the same without recourse to special system calls or memory models when a data object becomes larger than a hardware-imposed limit. However, developers must be willing to accept that end users are unable to run the application on a 286-based PC. In contrast, OS/2 allows all applications to exploit efficiently a broader range of 286-based PCs. An OS/2 developer must take special efforts to build applications that are portable to Unix or other non-Intel platforms.

Disposable-memory services. In a virtual-memory system, total allocated memory can exceed physical memory. This is done by swapping some memory regions (pages in Unix, segments in OS/2) to disk, thereby freeing physical memory. When a process refers to an address in a region not in physical memory, the system suspends the process and reads the saved segments or pages back from disk. This is transparent to the process. Read-only segments, such as pure code, are *disposable memory*, meaning the memory can be reused without being written out to disk. OS/2 also allows the programmer to designate writable segments as disposable, thereby saving the overhead of writing them out.

Disposable memory is allocated by the normal DosAllocSeg or DosAllocHuge functions with appropriate option bits set. Before using such memory, the process must call the DosLockSeg function. If the segment had been discarded, the system returns an error. The process must request reallocation of the memory and then rebuild its contents. Thereafter, the segment is not a candidate for discarding until the process calls DosUnlockSeg.

Allocating disposable memory is another example of how OS/2 involves the programmer in management of system resources. Unix has no corresponding service, mainly because the page-swapping mechanism involves much smaller units than segments, making it impractical to discard segments on a page-by-page basis.

Memory locking. Both Unix and OS/2 allow some of a process's memory to be *locked*, or designated as physical-memory resident, so it will not be swapped to disk. Memory locking is useful when a critical process always must be able to execute and the time required for disk access is an unacceptable delay.

Memory locking is useful when a critical process always must be able to execute and time required for disk access is unacceptable.

Unix provides a kernel call, **plock**, that can be used both to lock and unlock the code and/or the data space of a process. It is not possible to lock only certain regions of code or private data into memory; the entire code or data space (or both) must be locked. However, shared memory segments can be locked individually through an option to the **shmctl** call. A process that invokes the **plock** call must possess the Unix superuser ID (designating full access to all system functions), so that this call cannot be used by an unprivileged user.

In OS/2, locking process memory spaces is done in the module definition file at link time. OS/2 has a finer degree of control than Unix because individual segments can be locked. However, the locks are static for the duration of the process. Dynamically allocated memory, such as shared-memory segments, cannot be locked, and it is not possible to lock a segment for only a short time during execution, such as the duration of a critical operation. This could be a drawback for realtime applications.

Memory aliasing. In protected-memory systems such as Unix and OS/2, code segments cannot be written and data segments cannot be executed. Although usually desirable, such strict protection sometimes is a hindrance—for example, when instruction code needs to be dynamically created and executed. Because segment-access characteristics are defined in segment descriptors, developers can bypass the restriction by creating two descriptors—one that defines a data segment and one that defines a code segment—as aliases for the same physical segment.

OS/2's DosCreateCSAlias function performs memory aliasing. Given a data-segment selector, it returns a selector for a code segment of the same size at the same physical address. The program can use the returned selector to construct pointers to executable code in that segment.

Although Unix has no directly equivalent kernel service, its **sysi86** call provides a pathway to obtaining such a service. However, **sysi86** directly manipulates the segment descriptors and therefore requires an understanding of the 386 segment-descriptor formats. The **sysi86** function is also nonportable; however, in this case, portability is not an issue because the purpose usually is to generate machine-specific executable code.

CONTROLLING PROCESS DESTINY

API task-control services allow application processes some measure of control over their own destinies, including how they are influenced by other processes and system elements. These task-control services also permit processes to call on other processes to perform work for them.

Unix and OS/2 offer rich sets of tasking services that, while they appear similar, have some distinct differences. Both systems allow a process to start another child process, but their approaches vary. In OS/2, a process conveniently can start a child process through a single system call, DosExecPgm, while Unix makes it more complex by requiring two system services, **fork** and **exec**.

The main difference between the Unix **exec** service and the OS/2 DosExecPgm service is that there is no return from the Unix call (it simply replaces the calling process with the child process), while there is a return from the OS/2 call. The OS/2 service is analogous to a call statement in a programming language, while the Unix service implements program chaining.

FIGURE 1: *Redirecting I/O of a Child Process*

UNIX VERSION

```
extern int errno;
int pfd[2], pid;
:
:
if (pipe(pfd) < 0)
    errexit("pipe open error", errno);
pid = fork();
if (pid < 0)
    errexit("fork error", errno);
if (pid == 0)              /* child process branch */
{
    close(0);              /* close std in */
    dup(pfd[0]);           /* dup read as std in */
    execl("readtest", (char *) 0); /* bring in new program */
}
                           /* parent process branch */
close(1);                  /* close std out */
dup(pfd[1]);               /* dup pipe write as std out */
:
:                          /* write to file desc 1, std out */
wait();                    /* wait for child to finish */
```

The fundamental differences between how Unix and OS/2 create child processes result in different handling of I/O hookups between parent and child processes. In Unix, which requires substantially less code, the child starts out as a copy of the parent; it inherits the parent's file resources unchanged, then modifies them before chaining to the target program. In OS/2, on the other hand, the parent starts the target program directly. Before creating the child process, the parent must save its resource configuration, then modify it into the configuration inherited by the child.

OS/2 VERSION

```
unsigned savin, savout;        /* variables for file handles */
unsigned piprd, pipwr;
unsigned newin, newout;
char missname[64];             /* message from DosExecPgm */
RESULTCODES getback;           /* structure for results   */
int err;
:
err = DosMakePipe(&piprd, &pipwr, 0);
if (err != 0)
    errexit("pipe open error", err);
savin = -1;                    /* use next available handle */
err = DosDupHandle(0, &savin);/* save std in */
if (err != 0)
    errexit("error saving std in", err);
newin = 0;                     /* dup pipe read handle as std in */
err = DosDupHandle(piprd, &newin);
if (err != 0)
    errexit("error duping pipe as std in", err);

err = DosExecPgm(missname, sizeof(missname),
                 EXEC_ASYNC,      /* constant for asynch execution */
                 0, 0,            /* no args, parent's environment */
                 &getback,        /* structure for results */
                 "readtest.exe");/* program for child process */
if (err != 0)
    errexit("exec error", err);
err = DosDupHandle(savin, &newin);  /* restore original std in */
if (err != 0)
    errexit("error restoring std in", err);
savout = -1;
err = DosDupHandle(1, &savout);     /* save std out */
if (err != 0)
    errexit("error saving std out", err);
newout = 1;
err = DosDupHandle(pipwr, &newout); /* dup pipe as std out */
if (err != 0)
    errexit("error duping pipe as std out", err);
:
:                                   /* write to std output */
wait();                             /* wait for child to finish */
```

If a Unix process has no reason to exist once its child is started, then the **exec** service is all that is required. However, if the parent process wishes to be active or suspended until the child terminates or otherwise signals the parent, then both **fork** and **exec** are required.

The Unix **fork** call creates a duplicate of the calling process, with a shared code segment and a duplicated data segment. To the parent process, **fork** returns the process ID of the newly created child; to the child process, it returns a value of zero. Each process can test the return value to take a different path of execution; if the purpose of the **fork** call is to start another process, the child eventually issues an **exec** call that chains the child process to the new program.

This two-step approach provides the child process with a full copy of the parent's environment (memory and I/O resources) that the child can modify before starting up the new process. Changes made by the child do not affect the parent. In OS/2, by contrast, the parent process must set up the environment for the new program and pass it to the child via arguments and flags in the DosExecPgm call.

The difference is especially significant in establishing I/O hookups between parent and child, as shown in figure 1. The code fragments show the creation of a child process that reads input piped from the parent. In both Unix and OS/2, the parent opens the pipe. In Unix, the child merely closes standard input, duplicates the pipe, then calls **exec**; the parent's standard input is undisturbed. In OS/2, the parent saves its standard input handle by duplicating it in another handle, switches standard input to the pipe, creates the child with DosExecPgm, then restores its own standard input.

The two systems differ further in executing child processes. The Unix **exec** service has six variations, each providing a different way to pass arguments to a child. OS/2 provides only one way at the system level but provides multiple capabilities in the **spawn** family of C-library functions.

Execution modes for child processes. Both Unix and OS/2 can achieve either synchronous or asynchronous execution of a child process, again going about it very differently. When a process executes a synchronous child process, the parent is suspended until the child either provides a signal or terminates. The parent then resumes, perhaps retrieving returned codes and/or data left behind by the child. Asynchronous execution of a child, on the other hand, allows the parent to continue execution in parallel with the child. Signals can alert the parent to significant child events such as termination.

OS/2 handles parent-child synchronization within the DosExecPgm function. The parent specifies synchronization options through a set of flags passed as one of the arguments to the function. Achieving similar results in Unix requires appropriate programming in the parent and child branches, following a **fork** call.

To achieve asynchronous execution, an OS/2 process sets an appropriate flag in one of the arguments to

DosExecPgm; the call then returns immediately after starting the child process. In Unix, the **fork** call always returns, so parent and child processes execute asynchronously by default.

The **wait** call in Unix and the DosCWait function in OS/2 provide a means for the parent process to wait for the child to terminate and to receive result codes. A terminating child process returns to its parent a one-word value in Unix and a two-word result structure in OS/2.

If an OS/2 process calls DosExecPgm with the synchronous flag set, the call does not return, thereby suspending the parent process until the child process terminates. Upon return, the parent receives the same result structure as from DosCWait. In Unix, the equivalent result is obtained by calling **wait** in the parent branch immediately after the **fork** call.

Both systems also provide for creating child processes in a debugging mode, in which the parent process acts as a debugger. In Unix, the parent performs a **fork**, a **wait**, and a **ptrace**, while the child performs a **ptrace** and an **exec** before debugging can ensue. Using the **ptrace** call, the parent process can examine or modify the target's memory and machine registers. In particular, Unix on the 386 permits the debugger to write to the debugging registers of the 386.

OS/2 implements debugging services in a similar way. The parent loads a process to be debugged by calling DosExecPgm with the EXEC_TRACE flag set; it then examines or modifies the child's code and data segments and controls its execution by calls to DosPTrace.

In general, both Unix and OS/2 provide reasonable debugging services. Both are constrained to debugging only captive processes, meaning neither system can capture a process on the fly and place it into a tracing state. The OS/2 facilities are easier to use because DosPTrace provides a richer repertoire of commands and services than does **ptrace**. Developing a multiprocess multithread debugger, however, is onerous in either system.

Threaded execution. Threaded execution is an abrupt point of departure for Unix and OS/2. In Unix, the fundamental scheduling entity is the process, while in OS/2, it is the thread. OS/2 resources such as memory segments and files are seen by the operating system as belonging to a process, but system time is divided among threads of all processes in the system.

OS/2 supports multithreaded execution with a set of process-coordination services for creating, suspending, and resuming threads—Unix provides no equivalent. OS/2 applications that make liberal use of multithreading facilities not only access unique services, but also can differ from an equivalent Unix application in ways that go as deep as the architecture. Applications that use OS/2's multithreading capabilities are not likely to be easily ported to the Unix environment.

The advantages of the OS/2 multithreaded design are primarily ease of development and some lowered system overhead for thread operations. Implementing the various functions of an application as multiple threads in a single program, instead of as separate processes, provides a single point of maintenance for procedures that are to execute in parallel.

Applications that make liberal use of the multithreading capabilities of OS/2 cannot easily be ported to the Unix environment.

Often, the design of a system points to a need for small asynchronous processing elements that install and execute quickly and conveniently. Individual elements might not be broad enough in scope to justify the overhead of a new process and complexities that arise when processes must communicate with each other. Additionally, they may be so tightly coupled to the specific requirements of the application that their utility to other applications is minimal.

Threads also have drawbacks; the most important is that none of the threads in a process is protected from another. Bypassing all the benefits of a protected-memory environment is possible through use of multiple threads. Bugs in an application, such as threads writing through invalid pointers, can be just as troublesome as in a system that provides no memory protection at all. Even though these bugs may not crash the entire system, they can still crash an application.

Extreme caution is necessary in the design and implementation of such applications to ensure proper coordination among threads as they compete for shared resources of the process. Because all threads in a process implicitly share its resources, access to data used by multiple threads must be controlled through serializing mechanisms such as semaphores or critical sections. Code that is developed under Unix or DOS may not work in an OS/2 multithreaded environment because it is not reentrant.

In Unix, the **fork** function provides some of the benefits of the OS/2 multithreaded processes. Given that **fork** creates a duplicate of the executing process and permits the child to take a different branch than the parent, a parent could start parallel paths of execution, all contained within the body of the parent process. Because Unix creates only a new instance of the process's data (code can be shared), this method is fairly efficient, but it requires special considerations for creating shared data segments that can be accessed from each of the paths.

Priorities and scheduling. Both Unix and OS/2 provide services that permit processes to have some say about how they are scheduled. The most fundamental way a process can exercise control over its own use of the system's time is by changing its own priority. The reasons that a process may need to change its priority vary. For example, some processes are *background* in nature, meaning that they are not time-critical. Lowering their priority means they execute only when other processes either have completed or are suspended while waiting for access to resources. Raising a process's priority is a more complicated issue.

Unix's multiuser nature and its scheduling philosophy, which attempts to give each user a fair share of the system's time, beget a hands-off approach to priority-adjustment services. Processes can have their priorities raised only by the superuser. Unix programmers perform priority adjustment using the **nice** system call to adjust the *nice value* of a process. The nice value is not the process priority, but rather a value to scale the process's priority.

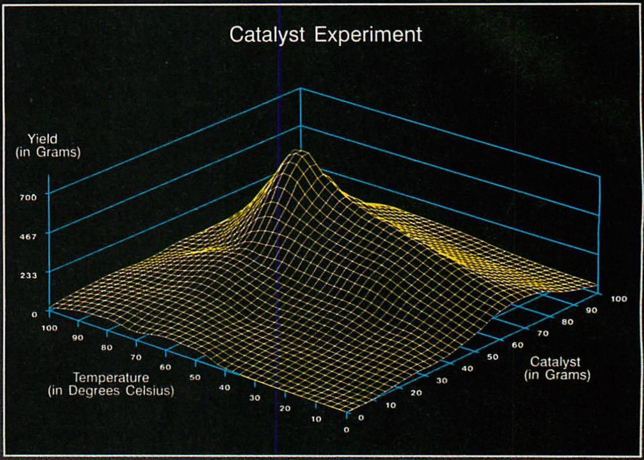
In OS/2, all priorities are assigned to threads. Threads have a freer hand to fine-tune their own priorities and those of other threads than do processes in Unix. The system assigns priorities in three dispatching classes: *time-critical*, *regular*, and *idle-time*. Within each of the dispatching classes, a thread can have a priority that ranges from 0 to 31. At each scheduling interval, OS/2 activates the highest-priority

The SAS® System

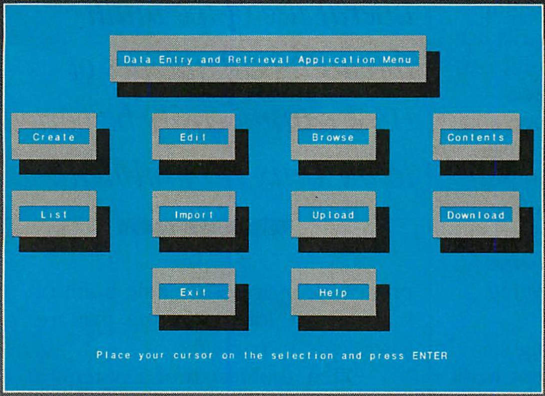
Powerful.

The SAS System brings power-hungry PC users efficient data management, reliable statistics, customized presentation graphics, and more. So you won't need separate packages for such tasks as market research, sales tracking, clinical trials, corporate reporting, and inventory control.

You can even create your own front-end menus. First-time users can analyze, report, and display data by simply filling in the blanks.



Productive.



The SAS System has integrated applications to use "as is" or customize to fit your needs. Plus a built-in micro-to-host link.

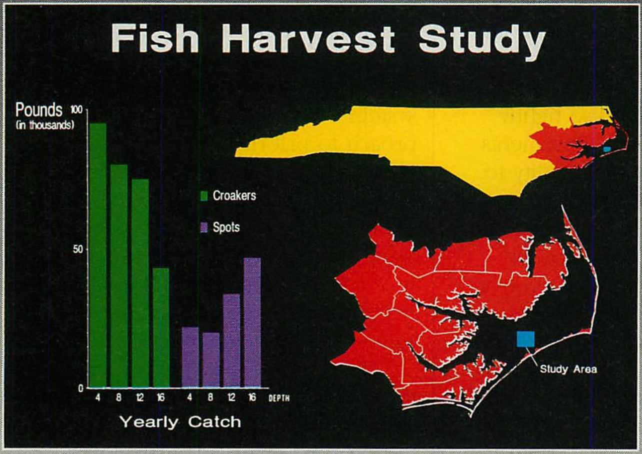
Read data in any format from any file, and manipulate data with common-sense statements like SORT and MERGE. Take advantage of superior analytical tools—from simple frequency counts to advanced regression. Then report your findings in tables, charts, maps, and more.

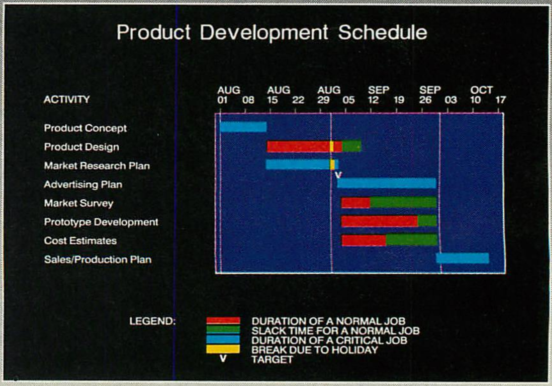
Prove it on Your PC.

The SAS System is already the software of choice for mainframes and minicomputers. Now this strength and flexibility can be yours on a PC...complete with on-line help, automatic updates, technical support, and high-quality training. Available only from SAS Institute, the leader in data analysis software since 1976.

Try the SAS System free for 30 days. For a closer look, call or write us today.

SAS Institute Inc.
Software Sales Department
SAS Circle ☐ Box 8000
Cary, NC 27512-8000
Phone (919) 467-8000 ☐ Fax (919) 469-3737





The SAS System runs under MS-DOS,® PC DOS, HP-UX, OS, CMS, VSE, VMS,™ PRIMOS,® and AOS/VS.

thread within the highest-priority class that is ready to run. If several candidate threads of the same class and priority are ready, the system starts them in round-robin fashion.

The first thread of a newly created process receives priority 15 in the regular dispatching class. A newly created thread inherits the class and priority of its creator but can change both by calling DosSetPrty. The change can be for the calling thread only, for all threads of the current process, or for all threads in the current and all descendant processes.

The most significant difference between Unix and OS/2, with regard to scheduling and priority manipulation, is the notion of the dispatching class. Even though the average programmer is best advised not to play with priorities, some developers need the flexibility in manipulating priorities offered by OS/2. Two types of applications that could benefit from these facilities are database servers, which must provide for a high volume of transactions over networks, and realtime applications. The OS/2 facilities allow more deterministic scheduling of critical threads and offer more promise to developers of such applications.

PROCESS GIVE AND TAKE

An operating system that supports multitasking systems must implement two conflicting characteristics. The first is memory protection, so that no process can inadvertently affect another. The second is interprocess communications, so that a process *can* affect another, but under controlled circumstances and only by design. Unix and OS/2 succeed on both counts. Memory protection is provided by the hardware and is essentially equivalent in both systems. Communications, implemented by software in each kernel, are similar in type, if not in execution. Both systems provide three major communications mechanisms: shared memory, queues, and pipes.

Memory sharing. When processes need to communicate among each other, the fastest way to pass information is for one process to write data into memory and for other processes to read it out of memory. The disadvantage is that this can defeat the memory protection that the system ordinarily enforces among processes. It takes some tricky programming maneuvers to open the door wide enough for cooperating processes to share memory, but not so wide that memory protection becomes a sham. No matter how good the design of an operating system, programmers must solve some memory-sharing problems on their own.

Apart from the differences arising from the memory architectures of underlying hardware, the base set of shared-memory services is similar in Unix and OS/2. Both systems provide services for identifying shared-memory segments by name. In Unix, the names are integer values called *keys*. In OS/2, the names are constructed according to the syntax of file names, beginning with \SHAREMEM\. In general, the OS/2 file-name syntax for naming shared memory (as well as semaphores and queues) is closer to a natural language than the integer identifiers of Unix, and it makes it less likely that two independently developed applications could use the same IDs.

The biggest difference between Unix and OS/2 with regard to scheduling and priority manipulation is the idea of the dispatching class.

In both systems, once a process allocates a named shared segment, any other process that knows the name can gain access to it. The creating process does not need to know who will use the memory. OS/2 provides two additional memory-sharing approaches for use between processes that know about each other (for example, parent and child processes). In the first approach, if a process sets the SEG_GIVEABLE flag when allocating unnamed memory with DosAllocSeg, it can subsequently call DosGiveSeg to allow another process to access that memory. The calling process must know the ID of the process to which it is giving the segment. The other approach is to set the SEG_GETTABLE flag when allocating unnamed memory, which allows another process to request access to that segment. The requesting process must know the selector used by the process that originally allocated the memory. (For more details, see Tech Notebook, this issue, p. 129.)

Because processes that share memory execute asynchronously, the potential exists for incorrect results to be passed if one process reads data that another process has only partially written into a shared segment. Processes must coordinate access to shared memory by means of a synchronization mechanism such as semaphores.

Semaphore services. Semaphores are flags that indicate one process's ownership of a common resource. Operating systems typically offer semaphores to solve two problems that exist in multiprogramming environments: serialization of access among multiple processes that want to access a common resource, such as a file or a shared-memory region, and event synchronization among processes (notification to a process that another process has completed some action).

Processes that cooperate through semaphores must request ownership of (acquire) the semaphore before accessing a resource or proceeding with a task dependent on another process. They also must relinquish ownership of (release) the semaphore upon completing access to the resource.

If a semaphore is already owned by another process and supports blocking (waiting, rather than immediately returning), then the system halts execution of the requesting process, places it in a waiting state, increments the count of processes waiting on the semaphore, and then schedules some other process to run. If a semaphore that is owned by another process does not support blocking, or if the calling process requests not to be blocked, the system returns to the calling process with an error code that indicates the semaphore is already owned and currently unavailable.

If a semaphore is not owned by another process, it is given to the calling process and the system returns with a code indicating that no error occurred. The calling process may take this to mean that it is now safe to proceed in its access to the shared resource, or that the action for which it was waiting has occurred.

Semaphore operations in Unix and OS/2 are similar only in the most general sense. The Unix services are really only the nuts and bolts required to implement semaphores; they cannot even make nested calls to acquire and release semaphores. This means that when a process acquires a semaphore, it must not execute code that subsequently acquires the semaphore again, because it will become deadlocked within itself waiting for the semaphore to be released.

On the other hand, the facilities provided by OS/2 permit the same process to acquire the same semaphore

The all-new Tandy 5000 MC Professional System is pure performance, from the Intel® 80386 processor operating at 20 MHz to the high-speed memory with cache controller that provides RAM-fast access to your data.

With the 5000 MC, you have the high-performance platform needed to take full advantage of industry-standard MS-DOS® applications, powerful new MS® OS/2 programs or multiuser SCO® XENIX® software.

Operating at 20 MHz, the 5000 MC cuts through the big jobs like database management, large spreadsheets and sophisticated graphics. The system architecture also provides a radical increase in data-transfer rates when the Tandy 5000 MC is configured within a 3Com® workgroup or multiuser environment. And its IBM® Micro Channel™ compatible architecture allows multiple processors to use the same bus.

The Tandy 5000 MC is the new alternative in personal computing.

Send me a 1989 RSC-20 computer catalog.

Mail to: Radio Shack, Dept. 89-A-825
300 One Tandy Center, Fort Worth, TX 76102

Name ____________

Company ____________

Address ____________

City ____________ State ____________

ZIP ____________ Phone ____________

Tandy Computers: Because there is no better value.™

Radio Shack
COMPUTER CENTERS
A DIVISION OF TANDY CORPORATION

Intel/Reg. TM Intel Corp. IBM/Reg. TM and Micro Channel/TM IBM Corp. MS, MS-DOS and XENIX/Reg. TM Microsoft Corp. SCO/Reg. TM The Santa Cruz Operation. 3Com/Reg. TM 3Com Corp.

CIRCLE NO. 243 ON READER SERVICE CARD

multiple times without becoming deadlocked. This feature of OS/2 permits program designers to use semaphores in a modular fashion within a set of subsystem services.

Another difference between the two systems is that Unix semaphores offer only two blocking options, *blocking* and *nonblocking*; the former can potentially wait forever. OS/2 offers an optional blocking time-out for processes that cannot afford to be suspended indefinitely.

Unix semaphores offer greater simplicity than those provided by OS/2. Unix requires only three system calls to create, delete, and manipulate semaphores, as opposed to the nine calls furnished with OS/2. However, this is a dubious benefit; developers in Unix may find they need to implement a custom set of functions for semaphore operations that invoke the Unix system calls indirectly. In addition, a standard for semaphore usage is unlikely to emerge if each developer must create custom functions.

Probably the greatest advantage to the Unix services is that they are more open to the developer who wishes to use them to create customized semaphore services. Unlike OS/2, Unix permits a process having the proper privilege to interrogate semaphores about the number of processes waiting for them, as well as the process ID of the last process to operate on the semaphore. Additionally, in Unix the process that created the semaphore can clear it so as to free all suspended processes and give them an error code, indicating occurrence of a semaphore error.

The intrinsics of both Unix and OS/2 can resolve most fundamental problems that occur with semaphores. These problems occur when a process that has acquired a semaphore dies a natural death while other processes are still waiting for the semaphore. Both operating systems provide for the semaphores to be released, allowing other processes to continue execution. Each system also allows for processes to be notified when a semaphor for which they are waiting has been deleted from the system.

The two systems have different means to identify semaphores globally. Unix creates semaphore sets, each identified by an integer key and containing one or more semaphores. The semaphore set is an array of structures; each semaphore within is identified by the handle to the set (returned by the **semget** call) and its ordinal index in the set. Processes other than the creating process can gain access to the semaphore by using the integer key to obtain the semaphore set handle.

OS/2 permits global (system) semaphores to have a name in a form similar to a file name, \SEM*name.ext.* Once the semaphore is created and named, it can be accessed through a handle analogous to a file handle. Other processes can access the semaphore by performing an open operation on the semaphore name. Both operating systems require that cooperating processes be written with the knowledge of semaphore identifiers.

In keeping with its multiuser heritage, Unix allows a developer to assign a controlled set of access permissions to a semaphore. Processes can create semaphores for exclusive use for themselves and their children or for global use by all processes. Processes can be granted access to get or to get and set semaphores, or they can have no access at all. OS/2 provides only two options: semaphores can be exclusive to the creating process and its descendants, or they can be nonexclusive, meaning that they can be globally accessed by any process.

In keeping with its multiuser heritage, Unix allows a developer to assign a controlled set of access permissions to a semaphore.

OS/2 provides a variation called a RAM semaphore that creates a semaphore structure in the data space of a process. This semaphore can be accessed only by the threads of that one process, but it provides for faster operations by eliminating some system overhead associated with a global semaphore known by a file-name identifier. Unix does not provide an equivalent mechanism. Even though semaphores in Unix can be private to the process that created them, they still incur the same system overhead as globally accessible semaphores.

Queue services. Queues are the most flexible of IPC facilities. They are byte streams written by one or more processes and read by a single process. The message queues and queue services in Unix and OS/2 are both similar and different.

In OS/2, queues are unidirectional. A process creates a queue using the DosCreateQueue service and assigns it a file-like name starting with \QUEUES\. The process that creates it is the owner of the queue; it can both write to and read from the queue. Any other process that knows the queue name can open it but can only write to it. The ID of the creating process is returned to each process that subsequently opens the queue.

OS/2 queues operate in one of three modes—first-in, first-out (FIFO), last-in, first-out (LIFO), or prioritized—selected via an option bit when the queue is created. The system places messages in the queue according to the mode: at the head of a FIFO queue, at the tail of a LIFO queue, or ordered by priority and arrival time.

The process that owns the queue can retrieve items either sequentially or randomly by item number. To find a nonsequential item to read, the process can scan the queue using the DosPeekQueue function, which returns the identifier of an entry in the queue and its contents but leaves the item in the queue. The DosReadQueue function performs the same function, but in addition, removes the specified item from the queue.

Any process that has opened the queue can use the DosQueryQueue service to view the status (number of items) of the queue. The owner can purge all items from the queue (without reading them) by calling DosPurgeQueue. The queue ceases to exist when closed by the owner or when the owner terminates.

A limitation of OS/2 queues is that the items they contain are not actual messages but pointers to messages. Each queue item consists of four values: the process ID of the writer, a one-word event ID, a message length, and a far pointer to a message. Communicating processes can use the event ID to pass any 16 bits of information. These processes also must separately arrange for the addressability of message data in the address space of the reading process; shared memory is the most convenient approach to this (for an example, see Tech Notebook, this issue, p. 129).

Unix message queues are completely bidirectional, and the queue items are not limited in length. Any process can both read and write a queue. The system copies the message from the sender's data space into a kernel buffer on writing, and from there into the reader's data space on reading.

When created, a Unix queue is assigned a handle. Any process that knows the key can get the handle by opening the queue; processes that know the handle can read and write the queue without opening it. Unlike OS/2, processes that open a queue are not made aware of the ID of the creating process. If this information is necessary, the developer must pass it to the reading process by some other mechanism.

The Unix message queue operates in FIFO mode only, and it cannot be read without removing the messages. A message type, associated with each message when added to the queue, is the first long word of the actual message buffer. Specifying a nonzero message-type parameter with the **msgrcv** call limits the reading of messages to that message type. This allows multiplexing messages on the same queue. For example, a server in a client/server relationship might be constructed to receive messages of type 1 only and then to respond to requesting clients using a different type, possibly equal to the client's process ID.

Unix message queues are established with the same permission attributes as files have in the Unix file system. Access to a queue, for example, can be restricted using either write-and-read or read-only access to processes other than the originating process or a process in the originator's group. The **msgctl** service provides status and control access to the queue, with access to the group ID, user ID, and permission attributes of the queue, as well as information on queue contents. Unix returns more status information than OS/2 does, including time of last send and receive, process IDs of the last transmitter and receiver, and the number of messages in the queue. A program also can flush or remove the queue using this service.

The differences between Unix and OS/2 queues are significant. The Unix approach is more powerful because of its automatic transferring of messages, bidirectionality, and better status reporting. OS/2 queues have an advantage in their nondestructive peek capability. With only one process able to read a queue in OS/2, the lack of message typing is not a serious drawback because this feature is typically used to route messages to different processes. For a single reader, similar functionality can be implemented by use of message priorities and of the DosPeekQueue service to scan the queue for messages of interest. However, the need to pass messages through shared memory makes OS/2 queues more difficult to use. This is another example of how OS/2 requires more involvement on the part of the developer in managing system resources.

Pipe services. In both Unix and OS/2, pipes are a form of IPC handled by a file I/O model. Most commonly, pipes direct the output of one program into the input of another. Command shells under both systems support the pipe notation; for example, **dir | sort** sends the output of the **dir** command as the input to the **sort** command, producing a sorted list of files. Both systems support *anonymous* (unnamed) and *named* pipes.

The implementation of anonymous pipes is functionally identical in Unix and OS/2. Such a pipe is created using the Unix **pipe** call or the OS/2 DosMakePipe function; each returns two file descriptors or handles—one for reading and one for writing. Thereafter, the pipe acts like a standard sequential data file, except that it does not have a movable file pointer. The only practical way of passing pipe handles to another process is by inheritance; therefore, only the parent process and its descendants can communicate by means of unnamed pipes.

Unix queues allow automatic transferring of messages, bidirectionality, and better status reporting; OS/2 queues are harder to use.

Similarity of pipes under the two systems might encourage development (or porting) of Unix tools that take advantage of pipes to provide a rich OS/2 development environment.

Named pipes in Unix and OS/2, on the other hand, have nothing in common. Named pipes in Unix, also called FIFOs, are created using the Unix **mknod** call, which also creates directories and other special files such as device-driver files. The name follows the syntax of file names, with no required prefix. Once created, FIFOs can be opened, read, written, and closed like a regular file, but access can be only sequential. FIFOs in Unix serve the same purpose as anonymous pipes; however, they can connect not only related processes, but any processes that know the pipe name. OS/2 provides no equivalent service.

OS/2 named pipes are currently available only with the Microsoft OS/2 LAN Manager, not as a part of the OS/2 kernel. They provide the basis for client-server relationships and distributed IPC over local area networks (LANs), instead of on only one machine. The server side of an OS/2 named pipe must use special pipe-oriented services for connecting, disconnecting, reading, and writing; the client side uses standard file services for opening, closing, and I/O. The closest Unix equivalents for communications over distributed systems are data structures called *sockets* and *streams*.

FILE SYSTEMS: BREEDS APART

A file system and its associated file I/O services provide a logical and consistent mechanism by which the kernel, the system, and applications access physical devices. The modularity and flexibility of the file system can afford significant capabilities with respect to system enhancement and expansion. The file-system services available in Unix and OS/2 are similar in that each uses a hierarchical directory structure. Although the underlying logical organization of disks in the two systems is different, the OS/2 and Unix file systems appear similar to an application.

The OS/2 file-system disk structure is identical to that of its ancestor, DOS. As in DOS, OS/2 can establish buffers to provide a limited form of disk caching. OS/2 disk caching, termed *sector buffering,* follows a least-recently-used (LRU) algorithm to free previously used buffers. The user can specify the number of sector buffers for use in the caching operation by means of an entry in the CONFIG.SYS file. Unix, on the other hand, includes a much more robust buffer-caching system that uses a multiple-hashed, free-list queue. The system configurer can establish the number of hashing queues and buffers per queue.

Unix can mount multiple volumes transparently under a single directory structure. The file-system services automatically traverse this structure, changing physical devices to access the appropriate file. This technique has been enhanced in systems such as Sun Microsystems' Network File System (NFS) and AT&T's Remote File System (RFS) to mount distributed, remote-directory structures together under a single directory tree. Under the OS/2 LAN Manager, a directory on a network device

can be mapped to a local-drive identifier, but there is no provision for combining directories on multiple devices into single trees.

File-access services are similar in Unix and OS/2. Both systems have the same set of calls to access true disk files, character I/O devices (serial ports, printers, tape drives), and IPC mechanisms such as pipes and FIFOs.

Both Unix and OS/2 establish a user file-descriptor table for each process. Each entry in the table identifies the location of a file on disk; although the mechanisms differ, the differences are transparent to applications. In both systems, multiple descriptor-table entries and file-table entries can point to the same physical file. The number that serves as the index into the descriptor table is called a *descriptor* in Unix and a *handle* in OS/2.

Each system can create multiple pointers to one file in the descriptor table. For example, handles for pipes created by a parent process can be duplicated as the standard input and output descriptors of a child process. This allows the child to accept what would normally be its standard input stream from a pipe written by the parent, and to direct its standard output via the write pipe back to the parent or to another process.

In Unix, developers duplicate a descriptor by calling the **dup** function. It accepts a single parameter—the descriptor number of an existing file to be duplicated. The kernel points the lowest empty descriptor-table entry to the same file identified by the input parameter. The calling process has no direct control over the descriptor number where the duplicate is entered.

To reassign a child process's standard input, the parent must close standard input (descriptor 0), then call **dup**. Because the standard input descriptor is now the first unused one, it becomes the duplicate of the pipe. A similar procedure is used to redirect standard output, which is descriptor 1.

The equivalent OS/2 service, DosDupHandle, is more flexible than Unix's **dup** function. It accepts two input parameters: the handle to be duplicated and the handle to be made the duplicate. If the second parameter is −1, the kernel uses the lowest unused handle for the duplicate and passes back that handle value. Otherwise, the specified handle is the duplicate; if it is already open, the kernel first closes the associated file. Standard I/O is redirected with the single DosDupHandle call to the kernel.

I/O synchronization. Normally, when either Unix or OS/2 performs a standard read or write to a disk file, the requesting process is suspended until the I/O completes. The transfer can be simply to the cache buffer, so that the calling process is awakened almost immediately if a buffer is available and no actual disk activity occurs on that call. If the data are not in a read buffer, or if all write buffers are full, the calling task waits while a physical disk operation makes a buffer available.

Because OS/2 processes can have multiple concurrent threads, nonblocking asynchronous I/O operations are easily supported. The DosAsyncRead and DosAsyncWrite functions allow the calling process to continue processing immediately after the read or write call. OS/2 creates a subsidiary thread automatically to start a read or write operation and waits for it to complete. When the operation ends, the thread clears a semaphore and terminates. The calling process can perform other operations in parallel; upon reaching a point where it needs the results of the I/O operation, the process waits on the semaphore.

Both Unix and OS/2 have the same set of calls to access disk files, character I/O devices, and IPC mechanisms, such as pipes and FIFOs.

An OS/2 process can specify the WriteThrough option when it opens a file. In a DosWrite call, this option tells OS/2 to immediately perform a write to the file, rather than save the output in a cache buffer. This important security feature guards against file corruption caused by unforeseen crashes in the process or the system. In lieu of establishing write-through as an automatic option for every write to a file, the process can call the DosBufReset function, which forces an immediate write to disk of the cache buffers for a particular file.

Unix has no automatic write-through facility. The **sync** function requests a write of all memory images to disk, including not only data buffers in the cache but all file-system directory information. However, **sync** is limited in that it schedules the write to perform at the system's next opportunity, which may not be immediately. Furthermore, it is system-wide, not specific to a file or process.

File locking and sharing. Under OS/2, a process specifies the file-sharing mode when opening a file. The whole file can be locked against subsequent reading, writing, or both. Only the process that originally locks a file can unlock it, but the system automatically lifts locks when the file closes. OS/2 also supports record-level locking. A process can lock any range of contiguous bytes in an open file by specifying to the DosFileLocks function an offset and the length of the region to be locked.

Unix also supports locking at the file and record levels, using the **fcntl** service. *Read locking*, sometimes called shared locking, allows other processes to read but not write to a locked file. *Write locking*, also called exclusive locking, prevents both read and write access by any other process. *Mandatory locking* for a file is a permanent attribute set via bits in the directory entry; the system automatically locks the file for the duration of use by any process. *Advisory locking* places the responsibility for requesting file locking on the user, who can maintain that lock over multiple-process executions.

File-permission attributes in Unix and OS/2 reflect the differences between multiuser and single-user systems. Under Unix, the attributes specify which users (the owner, the owner's group, or all users) have which kind of access (read, write, or execute). Under OS/2, the attributes are the same as under DOS and specify how a single user may access the file: normal (allowing all types of access), read-only, or system/hidden (invisible to user-level file functions).

Device I/O. Both Unix and OS/2 can perform raw I/O to a disk volume without benefit of a file structure. Unix reserves a volume to a process that performs this type of I/O. OS/2 allows other processes to access files within the volume using standard file-system calls, unless the process that initiates the raw-device I/O issues special DosDevIOCTL commands to lock and unlock the volume.

Both operating systems perform I/O to character devices, such as serial ports and printers, by means of device drivers. In addition to default drivers supplied with each system, the user can load specialized ones. In Unix, drivers can be installed dynamically, while in OS/2, installing a driver requires rebooting the system.

In Unix, devices have names that follow file-name syntax, beginning with the special directory /dev/, so they can be listed with standard directory-listing commands. In OS/2, however, device names are outside the file space and no way exists to obtain a directory of OS/2 objects that have file-like names, such as devices, shared memory, semaphores, and queues.

File status. OS/2 provides a wealth of calls that can be used to return status information about a file. Some are redundant, providing a subset of information available from other calls. For example, the DosQFileInfo function returns the same information for a specific file as DosFind does for the next file that matches a wild-card pattern. The DosQFileMode function returns only the attribute word. Separate OS/2 services also obtain the status for a file handle and a file name.

Unix provides somewhat fewer services than OS/2 for accessing file status, but the same information as obtained in OS/2 is available. The Unix **access** call does not actually return file status, but compares permission attributes of a file with a pattern and determines whether the specified access is permissible. The **stat** call returns file status when passed a file name, while **fstat** and **fcntl** return information about a file, given its descriptor.

SHARED LIBRARIES

Because physical memory is a precious resource, Unix and OS/2 try to avoid loading multiple copies of the same data or code. The most obvious case, recognized and handled by both Unix and OS/2, is when the same program is being run multiple times. Although each activation of the program needs its own data, the program code can be shared by all processes using the application. This is especially important in a multiuser environment, where many users may be running the same application, such as an editor.

In a single-user environment, loading multiple copies of the same program is less frequent. It is common, however, to have many programs that use the same support libraries. For example, C programs written for the same C compiler are linked with a large subset of identical C-library routines. OS/2's dynamic link libraries (DLLs) and Unix's shared libraries are designed to reduce total system-memory needs by identifying and sharing common routines. In Unix System V, only release 3.0 or later supports the library-sharing feature.

Unix's shared libraries statically link with application modules that use them, so each shared library must be assigned a fixed address in memory. AT&T publishes a list of recommended locations for each type of library. This is a consequence of the traditional static link/load model for which Unix was designed, in which all addresses must be known either absolutely or relatively to the entry point at link time. Unix provides no inherent facilities within the kernel for dynamic binding of library code to application code. Calls to Unix kernel functions are resolved to stub routines in a statically linked library; these in turn access low-level functions through a machine-specific linkage.

In contrast, OS/2 provides a dynamic-link model: the loader can defer the locating of external references until load time (see "OS/2's Dynamic Link," Mary DeWolf and Ted Mirecki, September 1988, p. 100 and ".EXE Files, OS/2 Style," David A. Schmitt, November 1988, p. 76). This model is an intrinsic aspect of the OS/2 kernel, to the extent that API calls are dynamically linked.

Programs designed for both systems can avoid system-specific features by relying on the C library instead of direct system calls.

Unix's shared libraries can address only the instance data of the processes that use them. Shared libraries in OS/2, on the other hand, enjoy a more flexible data-space model. In addition to providing private instance data to using processes, the OS/2 DLL facility permits a shared library to possess a global data area. Using processes share access to this area. Access is controlled by the passing of pointers from the library routines back to the process routines that call them.

Although DLLs are one of the major strengths of OS/2, they suffer from two limitations. The first, hopefully temporary, is the inadequacy of OS/2's documentation—the necessary information is spread out over several volumes. The second drawback stems from the limitations of OS/2 as a system for a limited microprocessor—DLL code is especially sensitive to the memory-segmentation characteristics of the 286 and requires extra attention to source-level constructs and compile-time options.

COMMON GROUND

Despite their many differences, Unix and OS/2 are not totally incompatible operating systems, because the general types of services offered by each are similar. In many cases, porting a program from one system to the other is not difficult. Difficulties arise where certain features of each system have no counterpart in the other—for example, named pipes and bidirectional queues in Unix and multiple threads in OS/2. A program that uses these features will have to be redesigned before it can be ported.

Programs designed from the outset for both systems can best avoid system-specific features by relying on a library instead of on direct system calls. For example, the **malloc** function, available in all C libraries for both systems, avoids the differences in memory-allocation methods.

Where no direct equivalents exist in the libraries of the two systems, the developer can usually write customized functions to bridge the gap. For example, OS/2 C compilers provide several variations of the **spawn** function, while Unix compilers use **exec** calls for the same purpose. When writing a program to run under both Unix and OS/2, the easiest method might be to code the OS/2 calls and write customized **spawn** functions for the Unix version to call the appropriate combinations of **fork** and **exec** calls.

The choice of environment is rarely made on the merits of the operating system. Instead, the realities of the market drive the decision: What is the dominant system in the target market for this application?

For the scientific and technical community, Unix continues to be the most reasonable choice. However, if your application targets the mass-market world where DOS currently reigns, OS/2 provides the advanced operating-system services to support advanced applications. The commonality of high-level languages in the two systems allows you to cover both markets with minimum effort.

Robert R. Morris and William E. Brooks are among the founders of Applied System Technologies Inc. of Ft. Lauderdale, Florida. They have developed graphics user interfaces and distributed, realtime data-acquisition systems for industry.

Give this to and you hear from

Upgrading mass storage used to mean one interruption after the other.

Your users were very easily confused.

And you weren't always free to answer questions like, "How do I set up a subdirectory?" Or, "How do I upgrade the DOS on my hard disk?"

But, all that's changed. With Plus Hardcard™ 40 you can upgrade your users to 40MB easily and never have to worry about it again.

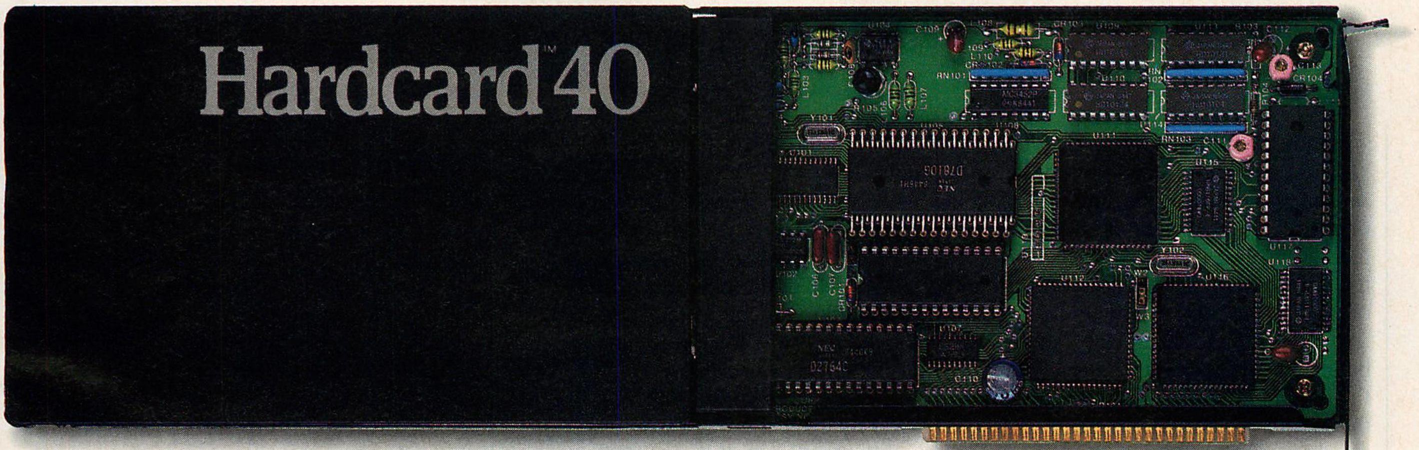
Because Plus Hardcard has been specifically designed with the end user in mind. It's completely different from other hard disks. For example, you won't have to personally install the Hardcard. Most of your users can do it in minutes. There's no system reconfiguration. No connector cables or adapter cards. And there's plenty of simple documentation ready to help those who don't know their keyboards from their clipboards.

Plus Technical Support Line
1-800-826-8022

There's even a toll-free technical support number for installation and user information. So if any of your

your users will never them again.

people have immediate questions, you won't have



to be the one with all the immediate answers.

More importantly, Plus Hardcard is the most reliable (40,000 hours MTBF) fixed disk available. It also happens to be extremely fast (access time 35ms) and durable (100 Gs). And it's compatible with some of the biggest names in the business. IBM.® COMPAQ.® Wyse.® And more. So if upgrade interruptions have kept you from getting to bigger issues, call us for complete compatibility and other technical information at 1-800-826-8022.

Plus®

Machines Corporation; COMPAQ/Compaq Computer Corporation; Wyse/Wyse Technology. © 1988 Plus Development Corp.

The Tenacious 286

Four new 286 computers—one from AST, one from Dell, and two from IBM—refuse to concede to the 386. They are small yet powerful machines that still have a constituency to serve.

DAVID CLAIBORNE

Although all but forgotten in the excitement of each new desktop development, the Intel 80286 is one old soldier that refuses even to fade away. The latest reincarnations are the compact 286-based desktop systems offered by AST, Dell, and IBM.

IBM first embraced the "smaller is better" movement with its PS/2 line. But the initial entry-level 286 machine, the Model 50, has always been criticized for its sluggish performance.

Since its introduction in April 1987, several companies have developed compact, entry-level 286 systems that offer better performance, lower cost, or both. AST Research is shipping the Premium Workstation/286, introduced at Comdex in November 1987, and Dell Computer unveiled its speedy System 220 last spring.

IBM has rejoined the fray with no less than two new 286-based systems. The company revamped the original Model 50 in June with a zero-wait-state (85-nanosecond) machine designated the 50Z, and turbocharged the original entry-level PS/2 Model 30 by replacing its 8086 with a 286. IBM announced the Model 30 286 in mid-September.

The price tags of these four machines reflect the new entry-level status of the 286, ranging from about $2,800 for the AST Workstation to almost $4,000 for the Model 50Z.

Make no mistake. Although powered by 286s, these are not simply small ATs. The Dell System 220 runs at a blazing 20 MHz with 70-nanosecond (ns) main-memory speed rivaling some 386-based systems. While not quite as nimble, the other machines operate at a respectable 10 MHz, 25 percent faster than the AT Model 339.

Aside from price and performance, these systems represent 286 desktop computing at its most refined. The manufacturers used application-specific integrated circuits (ASICs), surface-mount, and very large-scale integration (VLSI) technologies to produce compact, highly integrated machines.

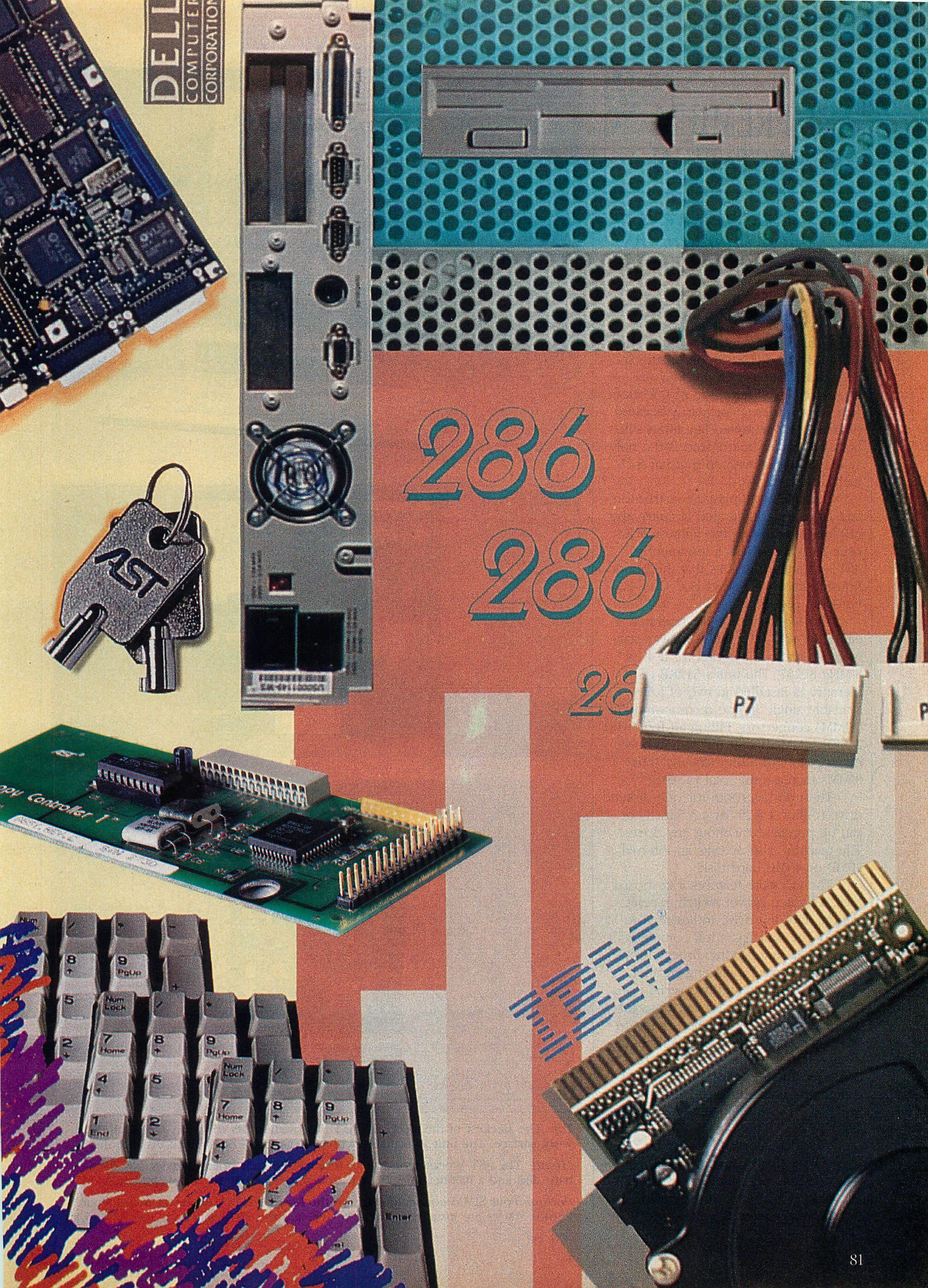
At a minimum, the system boards include serial and parallel ports, graphics and disk-drive controllers, and at least 2MB memory capacity. Total memory capacity is expandable to 16MB.

Hard-disk sizes range from the 20MB disk on the Model 30 286 to the optional 100MB drive available for the Dell 220. All four machines support AT software and, except for the 50Z with its Micro Channel bus, accept AT-class expansion boards. All support OS/2.

All of these functions are integrated into surprisingly small packages. The AST Workstation weighs in at just over 15 pounds and requires a scant 700 cubic inches of work space. In contrast, the original 1984-era IBM AT weighs over 40 pounds and occupies more than 2,000 cubic inches.

In short, the 286 desktop system has come a long way in four years. Today's system is small, powerful, and affordable. Whether destined for standalone use or connected to a LAN, the 286 has never had a better home.

COMPUTER GRAPHIC • NATALE/EAST


DELL
COMPUTER CORPORATION
AST
286
286
P7
IBM

AST PREMIUM WORKSTATION

When desktop space is at an utmost premium, AST has the answer.

The AST Premium Workstation/286, the smallest of the small machines (3 inches high by 16 inches wide by 14.4 inches deep), fits easily on a 24-inch-deep work surface. Of the four machines reviewed here, the AST Workstation is the quietest (primarily because of an extremely muted fan and a stiff, but clickless keyboard) and is the only system available with an internal 5.25-inch diskette drive.

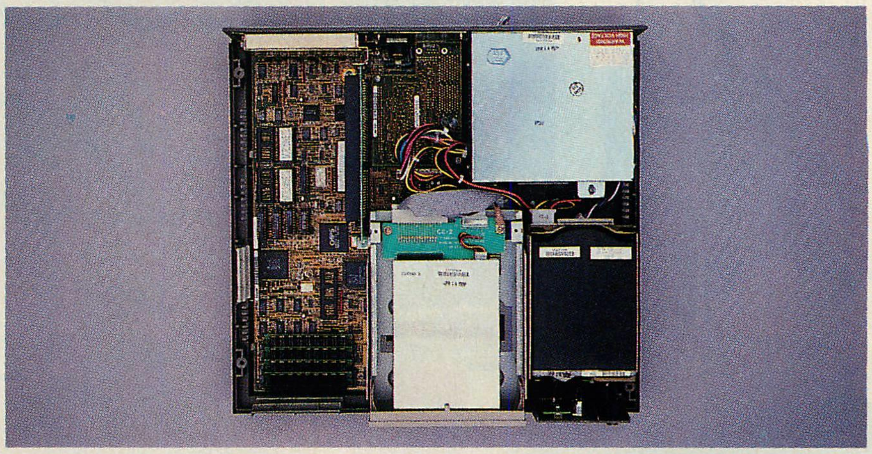
Because of all-plastic construction and large-scale integrated circuitry, the AST machine also is the lightest at just over 15 pounds. It is a testament to AST's engineering skills that the plastic housing meets the strict FCC Class B emission requirements.

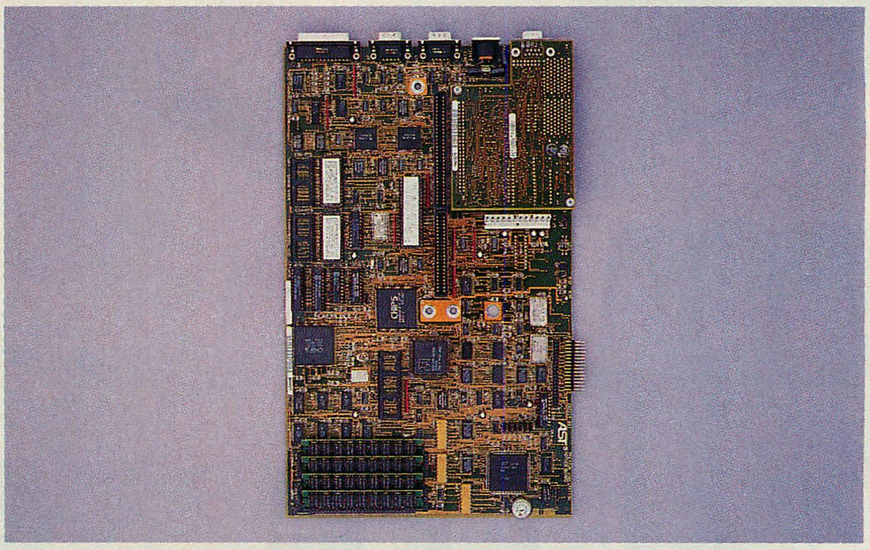
The Workstation's 286 operates at either 10 or 6 MHz with one-wait-state memory and has a socket for a 4- or 8-MHz 80287. The unit's 512KB of memory is installed in two of four available single in-line memory-module (SIMM) connectors. Filling all four with 256KB SIMMs brings the system-board memory to 1MB. Alternatively, 1MB SIMMs increase the total to 4MB.

The unit has room for one diskette drive (either a 5.25- or a 3.5-inch) or can be purchased without any drives. The machine also accepts an internal 20MB or 40MB hard disk.

The faceplate features a keyboard lock, a lighted power switch, a reset button, and disk-drive indicator. The back of the unit has a full complement of external connections directly from the system board including parallel, keyboard, display, and two serial ports (9-pin version), and slot openings for two expansion boards. The expansion bus is the standard AT bus, running at 8 MHz, and the keyboard is the 101-key enhanced model provided on all AST Premium computers (see "AST's Smart Machine," David Claiborne, August 1988, p. 92).

Video control is provided by means of a small daughterboard that plugs directly onto the system board and does not require an expansion slot. Monochrome/CGA, EGA, and VGA





Top: The sleekest of the sleek, the AST Workstation fits easily on a 24-inch deep work surface. The unit is available with either a 5.25- or 3.5-inch diskette drive.

Middle: The AST Workstation has an internal 3.5-inch storage device bay for a hard disk and a half-height 5.25-inch storage device bay for a diskette drive.

Bottom: Four SIMM connectors are on the lower left of the system board (four 256KB SIMMs are installed). The 286 CPU and the 287 socket are above them.

daughterboards are separately priced. The VGA version was not available at press time.

WHAT'S INSIDE?

Rather than the slide-off cover on most PCs, the entire top of the AST Workstation lifts off. Once inside, AST's engineering skill is immediately apparent. The system board is extremely small, despite its many capabilities.

The inside also shows the ubiquity of the Chips & Technologies (C&T) AT chip set. AST uses C&T ASICs for both the system board and graphics controller. All chips are surface mounted, except for processors, the ROMs, and the six AST programmable array logic (PAL) chips. AST manufactures its own PALs, putting the company in an engineering-development league with IBM and Compaq.

The diskette drive sits in the middle of the unit and the controller sits directly underneath, contained on a small board connected to the system board. The controller board, with its pin connection to the system board, is not up to the engineering standards of the rest of the machine, but is functional. To the right of the diskette drive is the Conner Peripherals hard disk and integrated controller, featuring 26 sectors, 4 heads, and 803 cylinders (42.7MB). Both drives snap in and out.

The 84-watt power supply (rear right-hand corner) is small but adequate for two expansion boards. Cables and connectors couple the power supply to the system board and drives, and the system board to the disk drives. Six of the system board's seven jumpers specify if the SIMMs are 256KB or 1MB (types cannot be mixed).

The system board has two other notable features. Three ferrite beads (FB1, FB2, and FB3) surround the keyboard connector to control electrical emissions. In addition, a group of circuit traces run from the video controller connector to a large blank area underneath the video controller board that is large enough to accommodate an integrated video controller.

Two horizontal slots for expansion boards using the XT or AT bus connect to the system board through a vertical connector. Although the slots use the standard AT bus, they allow only expansion boards that meet the XT height requirements. Taller AT-height boards work in the slots, but the system-unit cover will not fit over them.

Standard software includes MS-DOS 3.3 and a diskette of AST utilities (accessible directly or through the ASTMENU program) to set the configuration stored in CMOS memory, perform a useful system checkout, install the utility software programs, and configure the hard disk.

The setup program, ASTSETUP, also stored in system ROM, permits the full configuration of the system unit. Whenever the configuration sensed by the power-on self test (POST) differs from the configuration stored in CMOS memory, the program prompts the user to reconfigure the system, presenting a list of the current configuration values and options.

The user can access the ROM version of ASTSETUP by causing an error (such as by holding down a key when the system is booting) and then pressing Ctrl-Alt-Esc. The setup program also can be executed directly from DOS by entering the command ASTSETUP or using the ASTMENU program.

ASTMENU is used to run the setup program, initiate system diagnostics, install the AST utilities, or format the hard disk (either DOS or low level). ASTMENU contains straightforward menus; context-sensitive help screens make system setup easy for both new and experienced users.

The utility software also includes an expanded memory manager, RAM disk, print spooler, and disk-caching software. The first three utilities are from the AST RAMpage multifunction expansion board. The RAM disk and print spooler can be installed in either conventional or extended memory. The disk-caching program, licensed by AST from Multisoft, is a versatile program that uses conventional, expanded, or extended memory, although the best performance is achieved using conventional or expanded memory. Using extended memory requires the 286 to perform frequent time-consuming switches between real and protected modes.

Another program, SPEED.EXE, changes the operating speed of the 80286. Command-line parameters HIGH, LOW, and D specify 10 MHz, 6 MHz, or operation at the default speed specified with ASTSETUP (6 or 10 MHz). The CPU speed also can be increased by using a Crtl-Alt-up arrow key sequence and decreased by using a Ctrl-Alt-down arrow key sequence.

The Workstation comes with two slip-cased volumes of documentation plus a paperback covering MS-BASIC. One volume covers MS-DOS 3.3; the other volume is the user's reference, which is divided into two parts. The *Premium Workstation User's Manual* covers hardware, and the *AST Premium Utility Software User's Manual* describes the software contained on the utility diskette.

The hardware information provides instruction for setting up a new system and installing optional devices, such as additional expansion boards and more SIMMs for the memory board. Pin-out lists are provided for the external ports. The description of the utility programs is adequate for installing and using the software.

The information contained in the two volumes is sufficient for setting up and running the system and is equivalent to what has become the industry standard for documentation included with the computer. The information is not sufficient for such tasks as hardware checkout and maintenance or extensive debugging.

AST PREMIUM WORKSTATION/286 VITAL STATISTICS

Model 100X: $1,445
10-MHz Intel 80286 microprocessor
512KB memory
Intel 80287 coprocessor socket
Realtime clock
Two serial ports
Parallel port
101-key enhanced keyboard

Model 103X/105X: $1,495
All features of Model 100X plus
1.44MB 3.5-inch drive (103X) or
1.2MB 5.25-inch drive (105X)

Model 123X/125X: $2,145
All features of 103X/105X plus
20MB hard-disk drive

***Model 143X/145X: $2,495**
All features of 103X/105X plus
40MB hard-disk drive

Available Options:
512KB memory kit: $315
2MB memory kit: $1,324
AST monochrome/CGA video module: $145
AST monochrome display: $195
*AST EGA video module: $300
*AST EGA display: $695
AST VGA controller: $300
AST VGA display: $695
AST Ethernet adapter: $595
AST 3270/Coax II: $895
*MS-DOS/GW-BASIC 3.3: included in system purchase price until December 31, 1988, $85 thereafter
MS-OS/2 1.0: $325

An asterisk indicates the model reviewed and the options included.

DELL SYSTEM 220

This speed demon proves undoubtedly that 286s don't have to be slow.

The Dell System 220, the most ambitious machine produced by the Dell Computer Corporation, is twice as fast as the AST and IBM machines. The system's 20-MHz 286 microprocessor also runs at 6 MHz for AT compatibility. The chip is actually an 80C286 (a CMOS version manufactured by Harris Corporation under license from Intel), which reduces power consumption and heat produced. A 70-nanosecond memory makes possible zero-wait-state reads and one-wait-state writes at top speed.

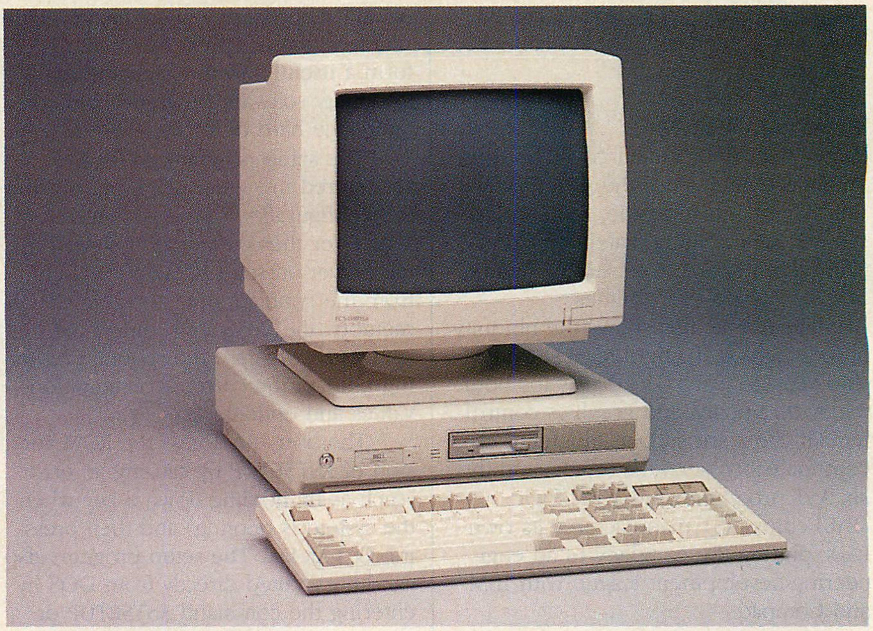
The 220's all-metal case (4.0 inches high by 15.0 inches wide by 15.7 inches deep) is smaller than the Model 50Z and larger than the AST Workstation and Model 30 286. The case plus the metal support cages inside make the unit heavy for its small size (22 pounds). The unit also has a loud cooling fan. The 101-key enhanced keyboard is made by Keytronics.

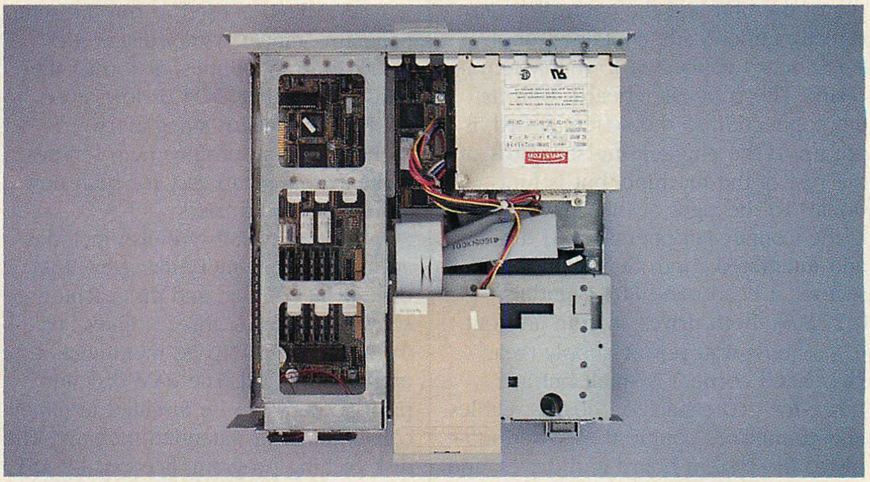
A socket is provided for an 80287, but only 8-MHz operation is supported, even though 10-MHz 80287s are available. The machine comes standard with 1MB of memory on the system board, contained in four 256KB SIMMs. Eight SIMM connectors allow system-board memory to expand to 8MB if 1MB SIMMs are used.

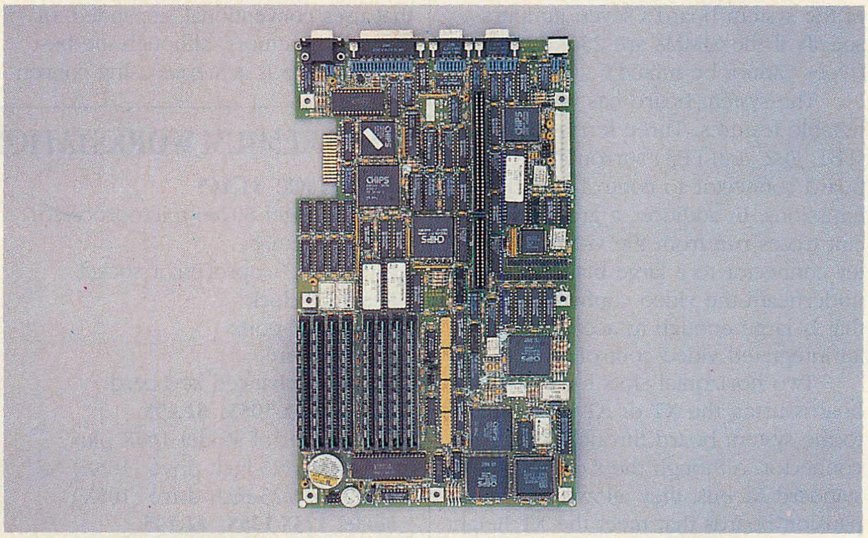
On systems with 1MB of memory, 640KB is conventional memory and 384KB is extended memory or holds a copy of system and video ROM. On systems with more than 1MB, this 384KB can be used only as fast ROM.

One 1.44MB 3.5-inch diskette drive is standard, but space is available to install a second drive if desired. Hard disks can be 40MB or 100MB.

The front of the 220 features a keyboard lock, power switch, and disk-drive indicator (but no reset button). The back of the system has a full complement of external connectors, including one parallel, one keyboard, and two serial ports, and a connector for the integrated VGA-display controller. The VGA controller supports the System 220's standard monochrome and optional VGA (.41mm dot pitch) and







Top: The Dell System 220 is just slightly larger than the AST Premium Workstation/286. The 220 can accommodate one or two 1.44MB 3.5-inch diskette drives.

Middle: The System 220's metal construction is evident throughout the system unit. The hard-disk drive is mounted beneath the two 3.5-inch diskette drives.

Bottom: The system board features eight SIMM connectors for a total capacity of 8MB. The optional 8-MHz 287 math coprocessor is located just below the SIMMs.

VGA Plus (.31mm dot pitch) color displays. The 220's dimensions and 100-watt power supply permit three expansion boards, mounted horizontally. Unlike the other systems reviewed here, the 220 can accommodate AT- as well as XT-height expansion boards.

AND OPENING IT UP . . .

The top of the Dell System 220 case is held firmly in place by six screws, three on each side along the bottom edge. Once the screws are removed, the top slides off easily.

The 220's internal layout is uncluttered, but not as elegant as the AST Workstation. The circuit-board work is about equal. Like AST, Dell uses Chips & Technologies (C&T) chip sets for both the AT logic and the integrated VGA controller. The Dell C&T chips, however, are all stamped 20 MHz.

Unlike the AST unit, the Dell VGA controller is integrated directly into the system board. Most chips are surface mounted, with the exception of the ROMs, processors, and PALs.

In contrast to the AST machine, the 220 has a profusion of jumpers scattered across the system board, 17 in all. These jumpers, not configuration programs, turn ports on and off.

The Dell 220 supports its three expansion slots and disk drives with metal cages. The expansion cage is welded to the bottom of the case, making access to the system-board components underneath difficult. The disk-drive cage supports two drives attached from the bottom; the entire cage has to be removed to install or remove the diskette drives or to gain access to the hard-disk drive.

The 220 features a Conner Peripherals diskette drive with an integrated controller. The standard 40MB hard disk has 17 sectors, 5 heads, and 975 cylinders for a capacity of 42.4MB.

Dell followed IBM's example of integrating the graphics controller onto the system board. A VGA feature connector is built into the edge of the 220's system board to support future VGA enhancements.

Standard software with the 220 is the System Support diskette, which contains the Dell System Analyzer and the Hard-disk Drive Setup programs. The analyzer is a complete series of programs to test the operation of the computer and all its peripherals and to isolate problems.

The setup program has two options, a low-level format or an automatic format, which partitions and formats the resident hard disk. Optional software includes Dell Enhanced MS-DOS 3.3 and Dell Enhanced MS-OS/2 Standard Edition with Dell enhancements, such as disk-caching and data-encryption software.

The primary Dell documentation consists of two manuals. The *Systems Support Manual*, a paperback guide to the programs on the Systems Support diskette, is supplied with all Dell computers. The explanations of the expected test results are good. The *System 220 Owner's Manual*, a spiral-bound paperback, has useful illustrations and instructions.

Although information in the two volumes is adequate for setting up and running the 220, it is not suitable for tasks such as hardware checkout and maintenance or extensive debugging. In addition, neither manual lists pinouts for external connectors. Supplemental manuals supplied with optional equipment provide this information. Separate short supplements explain the installation of options.

Because Dell has no dealers, it includes with each 220 a one-year on-site maintenance contract from Honeywell Bull Corporation.

DELL SYSTEM 220 VITAL STATISTICS

***Base Model: $2,299**

20-MHz Harris 80C286 processor
1MB memory
Intel 80387 coprocessor socket
Realtime clock
Integrated VGA controller
Two serial ports
Parallel port
1.44MB 3.5-inch diskette drive
101-key enhanced keyboard
VGA monochrome monitor

Available Options:

1MB memory kit (256KB-by-4 SIMMs): $995.95
VGA color monitor: $200[a]
*VGA Plus color monitor: $300[a]
*40MB hard-disk drive: $700[b]
100MB hard-disk drive: $1,500[b]
*80287 coprocessor (8 MHz): $400
Second 1.44MB 3.5-inch diskette drive: $200
MS-DOS/GW-BASIC 3.3: $120
MS-OS/2 1.0: $325

An asterisk indicates the model reviewed and the options included.

[a] *Additional cost if substituted for monochrome display.*

[b] *Additional cost if included in original purchase.*

CIRCLE NO. 227 ON READER SERVICE CARD

IBM PS/2 MODEL 50Z

Zero-wait-state memory comes to the Model 50.

The IBM PS/2 Model 50Z is part one of IBM's two-pronged attempt to recapture some of the market share the company lost since it introduced the PS/2 line. Part two is the newly remodeled Model 30 (described in the next section). With its zero-wait-state memory and faster hard disk, the 50Z is a significant improvement over the original Model 50, although the only external difference between the two is the nameplate.

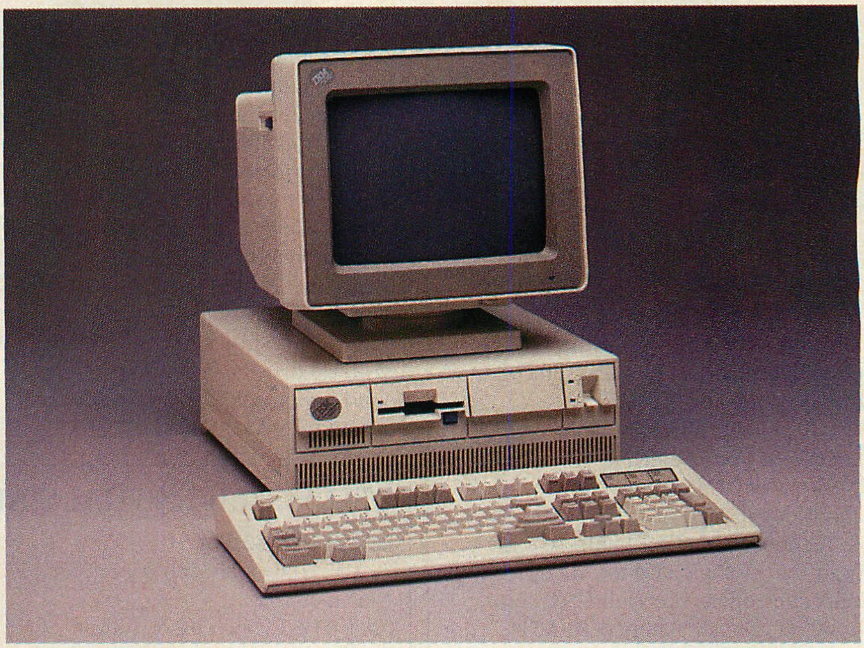
The 50Z's 286 runs at 10 MHz, and a socket is provided for a 10-MHz 80287, the fastest coprocessor available for the machines reviewed here. The system board contains 1MB of RAM in a single SIMM. The only way to expand system-board memory is to swap the 1MB SIMM for a 2MB SIMM. This is a $1,395 proposition, however, because the old 1MB SIMM is no longer useful. Memory beyond 2MB requires an expansion board. One 1.44MB 3.5-inch diskette drive and a 30MB hard disk are standard, with room for a second diskette drive. The 50Z is also available with a 60MB ESDI hard disk in place of the 30MB unit.

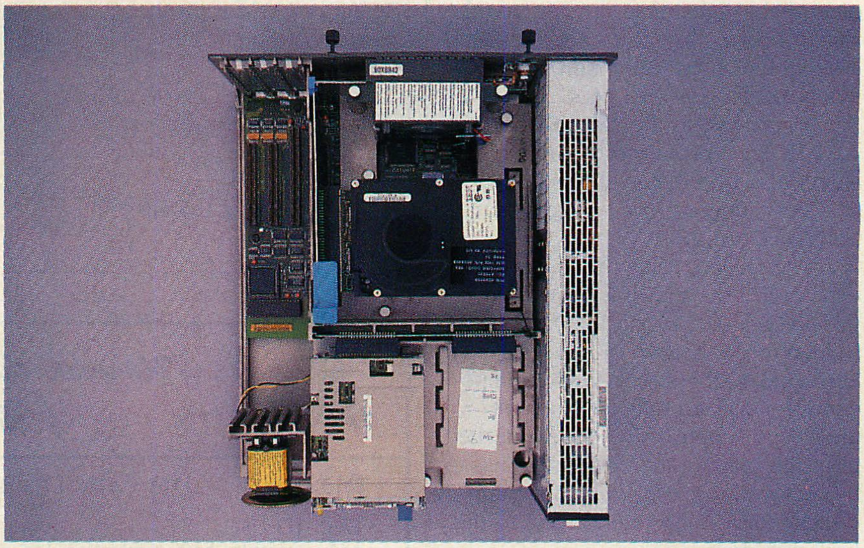
The 50Z's case (5.5 inches high by 14.1 inches wide by 16.5 inches deep) is higher and narrower than the AST and Dell machines primarily because IBM decided to continue using vertically mounted expansion boards. The standard IBM PS/2 desktop case, Lexan plastic with a metal cover, yields a unit that weighs 21 pounds.

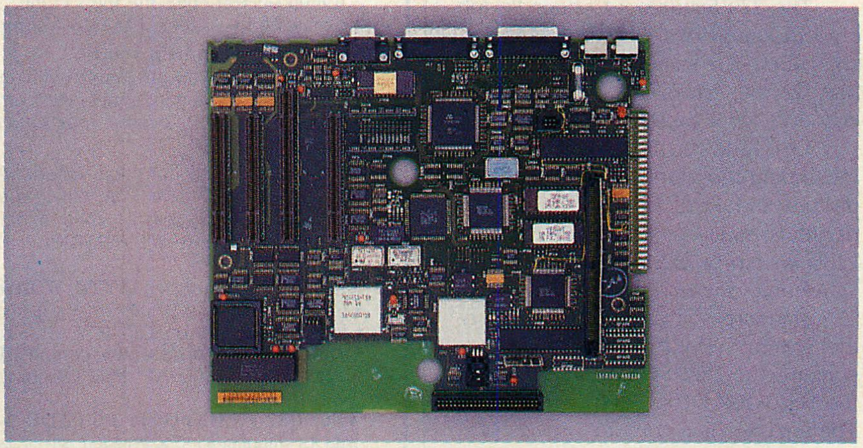
The power switch and disk-drive indicator are located on the front of the unit. The back features a keyboard lock, a 25-pin serial, parallel, keyboard, and auxiliary (for a mouse) ports, and a connector for the integrated VGA-display controller. The VGA controller supports all IBM PS/2 monochrome and color displays. A 94-watt power supply supports internal options and three Micro Channel expansion slots, which use a 16-bit Micro Channel bus. One slot has the IBM video extension.

INSIDE THE SYSTEM UNIT

Thumb screws make the 50Z the simplest of the units to open. Once inside, the first question is, where is the com-







Top: The IBM PS/2 Model 50Z, the largest of the units reviewed, uses the same case as the original Model 50. It does not fit on a 24-inch deep work surface.

Middle: The only cable to be seen connects the battery and speaker to the system board. The unit supports two 3.5-inch diskette drives and a 30- or 50MB hard disk.

Bottom: The Micro Channel connector on the far right is reserved for the hard-disk controller. The 286 CPU and 287 math coprocessor are at the lower left.

puter? IBM has significantly reduced the number of chips and the size of the system board. As a result, the system board occupies only the back two-thirds of the unit.

All of the components are surface mounted except for the ROMs, the processor chips, the SIMMs, and the unit's two discrete crystal oscillators used for VGA timing. The crystals for the processor and bus timing are integrated directly into the custom IBM chips.

As with the original 50, all the components literally snap in and out in a matter of minutes without tools, cables, jumper pins, or switches. Even the hard disk connects to the system board through a fourth Micro Channel connector. All configuration settings are stored in CMOS memory and changed by software programs.

The IBM Model 50/60 reference diskette, the standard software, contains a tutorial and a set of useful utility programs for configuring and testing. The configuration program is important because all configuration on PS/2 computers is performed with software. As the user adds expansion boards, the expansion-board configuration program must be copied to the reference diskette. The program to set up the IBM disk-caching device driver is included on the reference diskette as a hidden file. IBM PC-DOS and IBM OS/2 Standard Edition are optional.

The standard documentation is the *IBM Personal System/2 Model 50 Quick Reference.* Included in the documentation packet is a list of IBM technical manuals covering all hardware and software aspects of the PS/2 computers in general and the 50Z in particular. Although in many cases these manuals cost more than $100, at least IBM makes them available. AST and Dell do not have similar documentation.

IBM PS/2 MODEL 50Z VITAL STATISTICS

***Model 8550-031: $3,995**
10-MHz Intel 80286 microprocessor
1MB zero-wait-state memory
Intel 80287 coprocessor socket
Realtime clock
VGA
Serial port (25 pin)
Parallel port
Auxiliary port (mouse)
1.44MB 3.5-inch diskette drive
30MB hard-disk drive
101-key enhanced keyboard

Model 8550-061: $4,595
All features of 8550-031, except 60MB ESDI hard-disk drive instead of 30MB hard-disk drive

Available Options:
2MB, 85-ns memory module: $1,395
0-8MB memory-expansion board: $600
512KB, 120-ns memory-module kit: $215
2MB, 120-ns memory-module kit: $995
*80287 math coprocessor (10 MHz): $525
8503 monochrome monitor: $250
8512 color monitor: $595
*8513 color monitor: $695
Second 3.5-inch diskette drive: $245
Internal tape backup unit: $765
Token-Ring Network Adapter/II: $895
3270 connection board: $595
System 36/38 Workstation Adapter/A: $620
PS/2 mouse: $95
Internal tape backup program: $100
IBM PC-DOS/BASIC 3.3: $120
IBM PC-DOS 4.0: $150
IBM OS/2 Standard Edition 1.1: $325
IBM PS/2 Hardware Interface Technical Reference: $125

An asterisk indicates the model reviewed and the options included.

IBM PS/2 MODEL 30 286

Even the entry-level PS/2 Model 30 gets a 286.

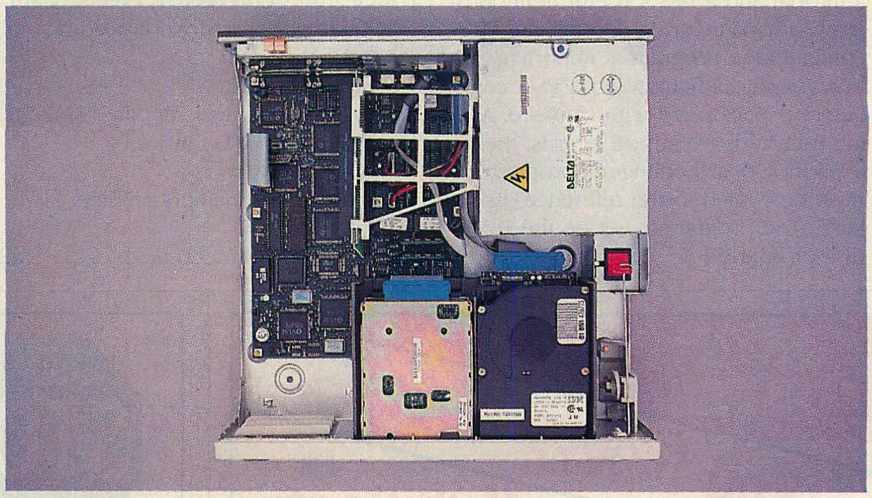
With a 286 microprocessor and a VGA controller, IBM has brought the entry-level PS/2 Model 30 into the OS/2 fold. (For a review of the original Model 30, see "Model 30: Apart from the Family," Ted Mirecki, August 1987, p. 92.)

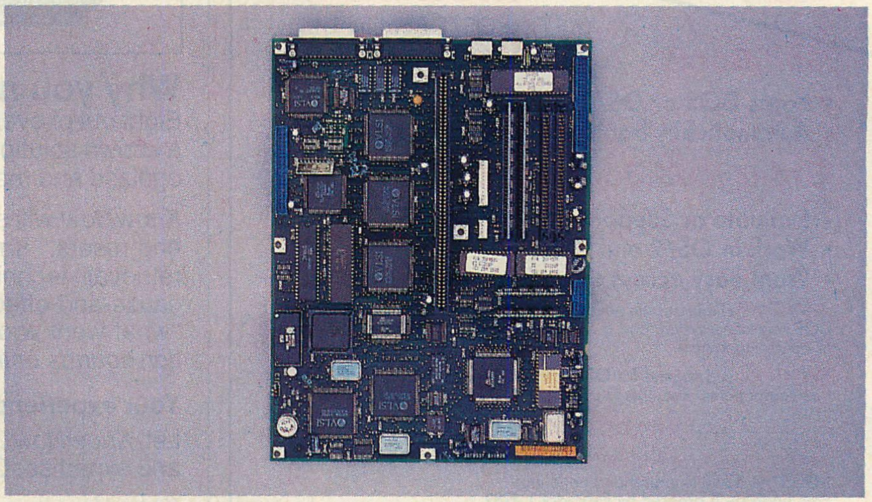
The only external difference between the original Model 30 and the Model 30 286 is the nameplate. The unit's height and width (4 inches high by 16 inches wide by 15.6 inches deep), similar to the AST and Dell computers, stem from its horizontal expansion slots. Even with a metal case (plastic is used for the front and back face plates) the Model 30 286 is still very light, weighing 19 pounds with hard disk installed.

The Model 30 286 is functionally an IBM AT running at 10 MHz. The system's 286 runs at 10 MHz and a socket is provided for an 80287 coprocessor, which runs at 6.67 MHz. The expansion bus operates at 10 MHz rather than the AT-standard 8 MHz. Model 50 features, such as fast microprocessor reset, advanced multitasking BIOS (ABIOS), and watchdog timer, are not provided. Two 256KB SIMMs on the system board bring the standard memory to 512KB. Four SIMM connectors allow 4MB of memory on the system board using 1MB SIMMs; additional memory requires an expansion board. RAM operates with one wait state inserted, ROM with two. The standard 1.44MB 3.5-inch diskette drive can be augmented with the optional 20MB hard disk, but IBM does not provide a cable for controlling a second diskette drive.

The optional hard-disk drive is the same 80-millisecond unit provided with the original Model 30. To better match the disk's performance to that of the 10-MHz 286, IBM has lowered the disk interleave from 3:1 to 2:1 and included disk-caching software with the Model 30 286.

The front of the unit features a power switch and disk-drive indicator lights, but no power indicator light. A lock on the right side of the cabinet secures the cover but does not disable the keyboard. The back side contains the standard complement of external





Top: The IBM PS/2 Model 30 286 uses the same case as the original Model 30. It supports one 1.44MB 3.5-inch diskette drive and one 20MB hard-disk drive.

Middle: The 30 286 has a larger power supply than the original 30, but otherwise it has the same layout, including the push-rod-connected external power switch.

Bottom: Four connectors on the right of the system board accommodate either 256KB or 1MB SIMMs. The 286 and optional 287 coprocessor are on the left.

connectors, including keyboard, auxiliary, parallel, and 25-pin serial ports, plus a connector for the integrated VGA display controller. The VGA controller supports all IBM PS/2 monochrome and color displays. A 90-watt power supply supports internal options and three AT expansion slots.

UNDER THE COVER

Removing the cover requires extracting two screws located on the each side along the bottom edge. The physical layout of the power supply, disk drives, expansion slots, and system board is similar to the original Model 30, but IBM completely revamped the system board using an AT-compatible chip set, consisting of six ASICs, from VLSI Technology. All system-board components are surface mounted except for the ROMs, the processor chips, the SIMMs, and crystal oscillators.

The Model 30 286 does not feature the level of internal engineering and snap-together construction found in the higher priced Models 50 and 70. Cables connect the power supply and the diskette drives to the system board. A ribbon cable runs under the entire system board to connect the diskette controller to the diskette drive.

The VGA controller requires a ribbon cable passing through a massive ferrite bead to span the distance from the connector at the back of the unit to the connector located in the middle of the system board. Unlike other IBM computers with a VGA, the Model 30 286 does not have a VGA feature connector, possibly limiting its use with future VGA enhancements.

The clock/calendar battery, which was mounted on the expansion-board connector in the original Model 30, is now soldered onto the system board. It is an extended-life, lithium battery that should last the life of the system.

Perhaps not so coincidentally, the Model 30 286 system board is identical in size to that of the Model 25 and uses the same connectors. In fact, when installed in an existing Model 25, the Model 30 286 system board works fine, indicating that a 286-based workstation from IBM for the collegiate set may be on the horizon.

The IBM Model 30 starter disk, the standard software with the Model 30 286, contains a set of utility programs that configure and test the computer. Expansion boards are still configured with their own switches and jumpers. The program to set up the IBM disk-caching device driver is included on the starter diskette as a hidden file. IBM PC-DOS and IBM OS/2 Standard Edition are optional. The Model 30 286 requires version 1.1 of IBM OS/2. Version 1.0 does not work. Version 1.1 is available to licensees of version 1.0 at no extra charge.

The standard documentation, the *IBM Personal System/2 Model 30 286 Guide to Operations*, explains how to operate the computer, install various options (80287 coprocessor, SIMMs), and perform limited troubleshooting. The information is adequate, but provides little technical detail. As with all IBM computers, additional documentation is available at additional cost.

IBM PS/2 MODEL 30 286 VITAL STATISTICS

Model 8530-E01: $1,995
10-MHz Intel 80286 microprocessor
512KB memory
Intel 80287 coprocessor socket
Realtime clock
VGA
Serial port (25 pin)
Parallel port
Auxiliary port (mouse)
1.44MB 3.5-inch diskette drive
101-key enhanced keyboard

***Model 8530-E21: $2,595**
All features of 8530-E01 plus 20MB hard-disk drive

Available Options:
512KB, 120-ns memory-module kit: $215
2MB, 120-ns memory-module kit: $995
0-12MB memory/multifunction board: $445
*80287 math coprocessor (10 MHz): $595
8503 monochrome monitor: $250
8512 color monitor: $595
*8513 color monitor: $695
Token-Ring Network Adapter/II: $895
3270 emulation adapter: $595
5250 emulation adapter: $497
PS/2 mouse: $95
IBM PC-DOS/BASIC 3.3: $120
IBM PC-DOS 4.0: $150
IBM OS/2 Standard Edition 1.1: $325
IBM PS/2 Hardware Interface Technical Reference: $125

An asterisk indicates the model reviewed and the options included.

FOUR FOR THE DESKTOP

Far from being obsolete, the hardy 286 thrives in these sleek, powerful AT-compatible machines.

All four computers are physically different from, but functionally compatible with, the traditional AT configuration (see table 1). With only a few exceptions, a variety of hardware and software products performed without problem (see table 2).

The AST Workstation and Model 30 286 are compatible with all the hardware and software tested, with one notable exception: the expansion bus allows only those expansion boards that meet the standard XT-height limitation. The AboveBoard 286 must be configured for 12-MHz operation to work properly with the Model 30 286's 10-MHz expansion bus. The Dell 220 worked with all hardware and software.

Because the Model 50Z does not have AT slots, it was not tested with expansion boards. All software, however, worked without problem.

The integrated VGA controllers of both the Dell and IBM systems were checked using *PC Tech Journal*'s VGA compatibility tests (see "The VGA Compatibility Test," Ed McNierney and Kent Quirk, November 1988, p. 48).

TABLE 1: ***Features Summary***

	AST PREMIUM WORKSTATION	DELL SYSTEM 220	IBM PS/2 MODEL 50Z	IBM PS/2 MODEL 30 286
PRICE (typical configuration)[a]	\$3,445	\$3,299	\$4,690	\$3,505
PROCESSORS				
CPU	80286	80C286	80286	80286
Coprocessor	80287	80287	80287	80287
Clock rate of CPU (MHz)	10/6	20/8	10	10
Clock rate of coprocessor (MHz)	4 or 8	8	10	6.67
MAIN MEMORY (16-bit data path)				
Base memory size (MB)	0.5	1	1	0.5
System-board capacity (MB)	4	8	2	4
Memory capacity	16	16	16	16
Memory speed (ns)	100	70	85	120
Wait states	1	0/1	0	1
DISK-DRIVE CONTROLLER				
Capacity (disk/diskette)	1/1	1/2	1/2	1/2
HARD DISK				
Capacity (MB)	20/40	40/100	30/60	20
Interleave	2:1	2:1	1:1	2:1
EXTERNAL STORAGE DEVICE BAYS				
5.25-inch (half height)	1	0	0	0
3.5-inch only	0	2	2	2
INTERNAL STORAGE DEVICE BAYS				
3.5-inch only	1	1	1	0
DISPLAY CONTROLLER				
Type	VGA	VGA	VGA	VGA
Data path (bits)	8	8	8	8
EXPANSION BUS				
Type	AT	AT	Micro Channel	AT
Speed (MHz)	8	8	10	10
AVAILABLE EXPANSION SLOTS				
16-bit	2	3	3	3
POWER SUPPLY				
Capacity (watts)	84	100	94	90
ELECTROMAGNETIC COMPATIBILITY				
FCC class	B	B	B	B
DIMENSIONS				
Width (inches)	16	15.1	14.1	16
Depth (inches)	14.8	15.6	16.5	15.6
Height (inches)	3.4	4.0	5.5	4.0
Weight (pounds)	15.2	22	21	17.2

[a] *The price is for a unit with one diskette drive, 20MB or larger hard disk, 1MB of memory, and EGA (or VGA, if available) controller and display.*

All systems feature a 286 microprocessor that runs at 10 MHz or faster. The System 220 features a 20-MHz 286 and is available with the largest hard-disk drive. The Model 30 286 has the slowest 287 math coprocessor and the smallest hard disk.

TABLE 2: *System Compatibility Summary*

	AST PREMIUM WORKSTATION	DELL SYSTEM 220	IBM PS/2 MODEL 50Z	IBM PS/2 MODEL 30 286
SOFTWARE				
Borland Lightning 1.01A	●	●	●	●
Borland SideKick 1.56A	●	●	●	●
Borland SuperKey 1.16A	●	●	●	●
Fifth Generation Fastback Plus 1.0	●	●	●	●
Living Videotext Ready! 1.0E	●	●	●	●
Microsoft Windows/286 2.03	●	●	●	●
Microsoft Word 4.0	●	●	●	●
HARDWARE				
Cheetah Systems Cheetah Card (with 2.5MB)	○[a]	●	N/A	○[a]
Hayes 1200B Smartmodem	●	●	N/A	●
Hayes 2400B Smartmodem	●	●	N/A	●
Intel AboveBoard 286 (with 2MB)	●	●	N/A	●
Microsoft Bus Mouse	●	●	N/A	●
Microsoft Serial Mouse	●	●	●	●

●=Yes ○=No N/A=Not applicable

[a] The Cheetah Card operates normally, but the system cover cannot be replaced.

All four systems demonstrate compatibility with commonly used software. However, the AST Premium Workstation/286 and PS/2 Model 30 286 cannot physically accommodate full-height, AT-type expansion boards such as the Cheetah Card.

TABLE 3: *VGA Compatibility Summary*

	DELL SYSTEM 220	IBM PS/2 MODEL 50Z	IBM PS/2 MODEL 30 286
BIOS			
Mode support	●	●	●
Cursor operation	●	●	●
Light-pen support	●	●	●
Multiple pages	●	●	●
Screen scrolling	●	●	●
Text I/O	○[a]	●	●
Graphics I/O	○[b]	●	●
Palette/DAC	●	●	●
Save/restore video state	●	●	●
Character generator	●	●	●
HARDWARE REGISTERS			
General	○[c]	●	●
DAC	●	●	●
Screen scrolling	●	●	●
Sequencer	●	●	●
CRT controller	○[d]	●	●
Graphics	●	●	●
Attribute	●	●	●

●=Yes ○=No N/A=Not applicable

[a] Incorrect color in Mode 11H.
[b] Palette reads and color page register calls work improperly.
[c] Input status register 0 not implemented.
[d] CRTC register values are different from IBM's; they may not be compatible with some VGA displays.

The Dell's integrated VGA is generally compatible, but color variations may be evident with some applications. For safety, the system should be tested with specific applications. The AST VGA module was not yet available for testing.

As expected, the Models 50Z and 30 286 VGAs were completely compatible with those of earlier PS/2s. The System 220's VGA also was generally compatible, a few inconsistencies occurred in operating the palette and color page registers (see table 3).

The *PC Tech Journal* system benchmarks were run on all four computers to measure their performance (for information on the benchmarks, see "High-level Measurements," Kent Quirk, September 1988, p. 54). Table 4 lists the test results.

The results verify the strengths and weaknesses of the four computers (see figure 1). All four are at least 25 percent faster than the 8-MHz AT. The AST Workstation and Model 30 286 exhibit about the same CPU and memory performance, and the Model 50Z is about 20 percent faster.

The 20-MHz processor in the Dell 220 excels in the HLSORT test, performing the test nearly twice as fast as the other machines. In the HLFLOAT test, however, Dell's speedy 286 is no advantage because the 287 coprocessor runs at only 8 MHz. Likewise, the Model 30 286, with its 287 running at only 6.67 MHz, performs the HLFLOAT test slower than all the rest.

The HLDISK results indicate that the disk drive in the Model 50Z is indeed faster than the earlier Model 50, but the drives in the AST and Dell machines are faster still. The Model 30 286, with its slow hard disk, cannot match the performance of the AT 339, with or without disk-caching software.

FORGOTTEN NO LONGER

All four machines provide substantially more computational power than the standard AT, and at much less cost than 386 computers.

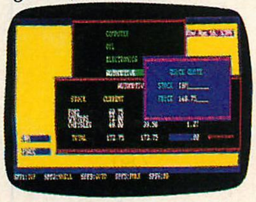
The AST Workstation is the smallest and has the lowest entry-level price ($1,495 in a single-diskette-drive configuration), but has lower performance than either the Model 50Z or System 220. The AST Workstation, like the Model 30 286, is handicapped by its one-wait-state memory.

The Dell's 20-MHz 286, coupled with 70-nanosecond memory, rivals the processing speed of more expensive 386 machines. If you are looking for speed and don't need the 32-bit capability of a 386, this is an excellent machine. Because Dell is a mail-order company, the only way to test a Dell computer is to buy one. Dell offers a 30-day, money-back guarantee and a standard one-year on-site maintenance plan to help ease anxieties.

How to match the best front end with the best back end.

You've invested in the best hardware, operating system and database products. You have a staff of programmers ready to go. But to build great looking applications with the sophisticated features today's users demand, you need a powerful front end development tool—JAM.™

JAM, the most advanced user interface management system on the market, does it all—from prototyping to implementation. And, because it works with any database or file manager, you get the best front end *and* the best back end.



Create colorful screens and nested windows easily.

JAM is hardware independent, so it isn't limited to one computer, database or operating system. In fact, JAM runs on everything from PC's to super-minis, works under 7 operating systems and provides access to a host of database products. Using JAM you can create a consistent user interface across multiple systems and hardware platforms.

JAM works under the following operating systems:

- UNIX®
- VMS®
- RMX™
- MS-DOS®
- XENIX®
- VOS™
- AOS/VS™

JAM makes it easier than ever to design and revise complex applications. Using features like context-sensitive help, procedural command language (JPL), shifting and scrolling fields, extensive data validations, on-line testing, and a variety of visual attributes, you'll be amazed how quickly your applications spring to life.

JAM is fast, as well as flexible. Development time is reduced significantly thanks to JAM's powerful screen drawing utilities and comprehensive library of subroutines. You can easily create and link together screens, windows, and menus to develop an application shell. Then simply attach the processing routines, and your application is complete.

Plus, if your back end is an SQL-compatible relational database like Oracle,® SQLbase,® Informix® or Britton Lee,™ you'll really appreciate JAM/DB*i*, JAM's optional database interface. With JAM/DB*i*, you can develop entire applications using only industry-standard SQL statements and JAM's authoring tools.

Use SQL to retrieve and display data.

So if you're programming in a 3GL like C or FORTRAN, follow the lead of the many Fortune 1000 companies who have already discovered JAM.

JAM from JYACC. It gives you the best at both ends.

Call for more information about JAM and our demo diskette. **800-458-3313**

JYACC, Inc.
116 John Street
New York, NY 10038
212-267-7722 FAX No. 212-608-6753

JYACC

JAM

JYACC Application Manager. *The Composer for Sophisticated Applications.*

CIRCLE NO. 128 ON READER SERVICE CARD

TABLE 4: *Benchmark Results*

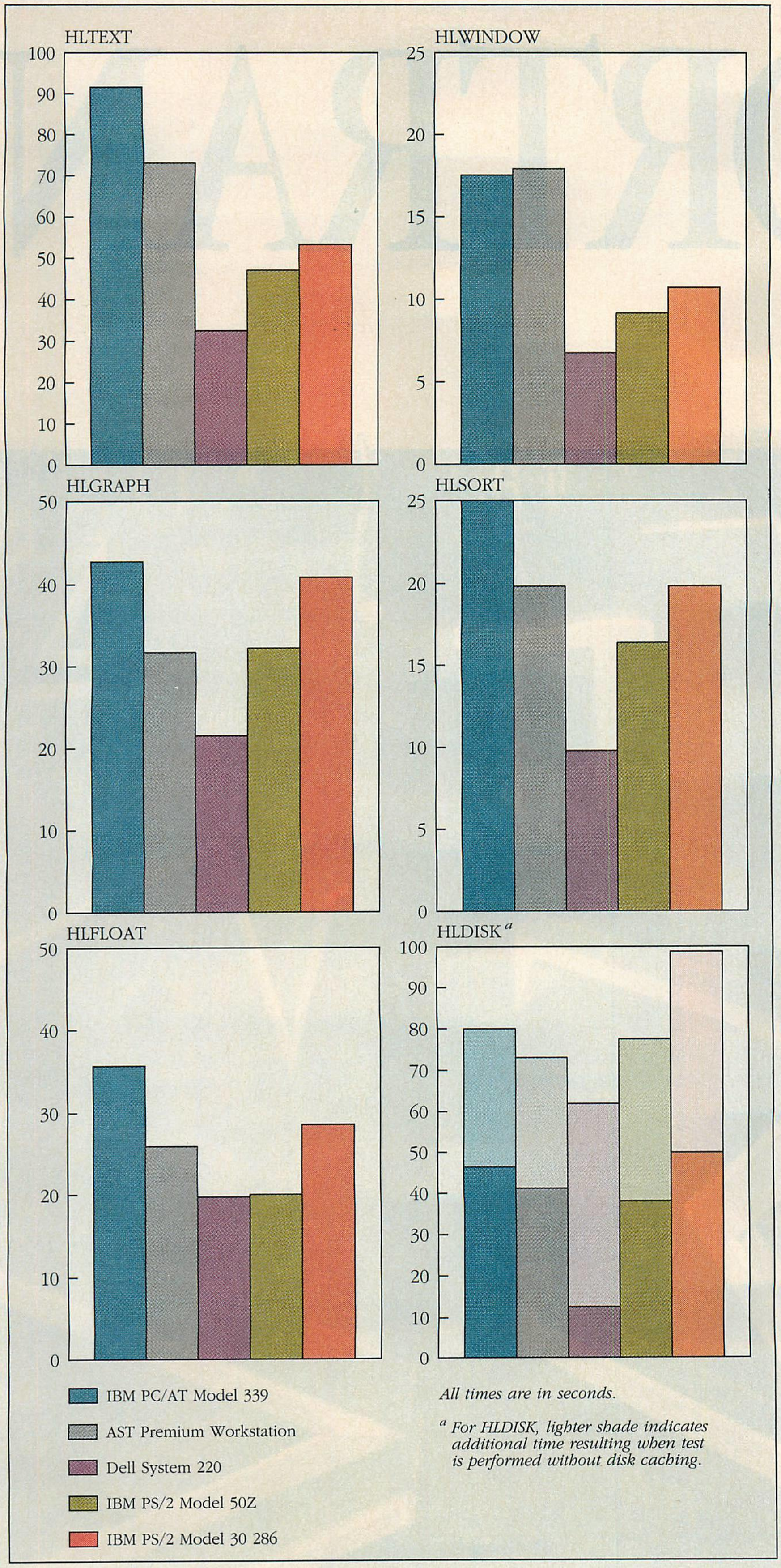
	IBM PC/AT MODEL 339	AST PREMIUM WORKSTATION	DELL SYSTEM 220	IBM PS/2 MODEL 50Z	IBM PS/2 MODEL 30 286
EQUIPMENT					
ROM BIOS date	11/15/85	03/25/88	01/15/88	04/18/88	08/25/88
Processor speed (MHz)	8	10	20	10	10
Coprocessor speed (MHz)	5.33	8	8	10	6.67
Base memory size (MB)	0.5	0.5	1	1	0.5
Video controller	8-bit EGA	8-bit EGA	8-bit VGA	8-bit VGA	8-bit VGA
Hard-disk size (MB)	30	40	40	30	20
HLTEXT (text scrolling)					
BIOS	26.37	18.46	8.68	12.58	14.06
DOS	29.67	20.87	9.67	14.94	16.92
C library	23.46	20.49	9.61	13.51	15.16
Windowed	12.08	13.24	4.50	5.98	7.08
Total	91.59	73.07	32.47	47.03	53.24
HLWINDOW (window/scrolling)					
Total	17.52	17.91	6.75	9.17	10.71
HLGRAPH (16-color graphics)[a]					
400 small areas	5.98	4.17	3.84	5.16	6.53
100 large areas	3.73	2.36	2.85	3.79	4.61
400 small ellipses	9.83	7.58	4.17	6.59	8.51
200 large ellipses	9.50	7.30	3.90	6.20	7.96
4,000 short lines	6.53	4.94	3.18	4.89	6.15
2,000 long lines	5.76	4.34	2.91	4.45	5.65
General graphs	1.37	0.98	0.65	1.09	1.37
Total	42.74	31.70	21.53	32.19	40.82
HLSORT (CPU/Memory)					
Data generation	2.08	1.64	0.82	1.37	1.64
Memory sort	23.02	18.18	8.95	15.00	18.18
Total	25.10	19.83	9.78	16.37	19.83
HLFLOAT (Fast Fourier Transform)					
Forward	18.13	13.13	10.05	10.16	14.50
Reverse	17.58	12.74	9.67	9.83	14.01
Total	35.71	25.87	19.72	20.00	28.51
HLDISK (with disk cache)[a]					
Data file creation	1.59	1.42	0.54	3.13	3.51
Index file creation	34.72	30.76	13.07	26.37	34.06
First report generation	3.62	3.40	1.48	2.14	2.80
Data reorganization	5.05	4.50	1.81	4.78	7.03
Second report generation	1.20	0.98	0.60	1.53	2.19
Total	46.37	41.15	17.52	38.02	49.89
HLDISK (without disk cache)					
Data file creation	3.40	3.29	3.29	3.18	3.51
Index file creation	44.67	37.80	26.75	36.59	43.07
First report generation	11.70	10.98	11.31	13.29	17.36
Data reorganization	16.64	17.47	17.14	20.98	31.92
Second report generation	3.35	3.35	3.29	3.29	3.90
Total	79.94	73.02	61.86	77.41	99.94

All times are in seconds, converted from 18.2-Hz timer ticks; therefore, total displayed is not always the exact sum of the individual results displayed.

[a] *The 256KB extended-memory disk cache was implemented using Multisoft's Super PC-Kwik disk cache on the IBM PC/AT; vendor-supplied disk-caching software was used with the other machines.*

The Dell System 220 is the best overall performer. The Model 30 286 does not fare well on the video or disk tests because of its slow ROM BIOS and slow hard disk. Its 8-MHz 287, however, makes its performance almost equal to that of the Model 50Z.

FIGURE 1: *Performance Comparison*



The Model 30 286's hard-disk drive is by far the slowest of the four machines reviewed; the disk-caching software included closes the gap, but not completely.

IBM's Model 50Z, a major improvement over the original 50, has the speed and power to compete with the AST and Dell machines, but is somewhat limited by its 2MB system-board memory capacity. The 50Z's primary advantage over its two competitors is that it is an IBM machine, with all the inherent quality and support that come with IBM products.

The Model 30 286 is the first AT compatible made by IBM. By using the Model 30 case, the VLSI chip set, and IBM's VGA circuitry, IBM has produced a machine that drastically lowers the entry-level price for OS/2 computing. Its performance, however, is also the lowest of the four machines reviewed.

The Model 30 286 is just that—a Model 30 with a 286. It is *not* a Model 50 with an AT bus. Like the AST and Dell machines, the Model 30 286 lacks the OS/2 support features, such as fast microprocessor reset and ABIOS, found on the Model 50Z and other Micro Channel-based PS/2 models. The Model 30 286 is competitively priced, but its limited disk capacity and performance make it better only in comparison to the original 8086-based Model 30.

Regardless of the manufacturer, all four of these machines represent a good value. The cost of reliable 286 computing has never been lower. Moreover, the sophisticated engineering and manufacturing techniques promise long-lived and reliable operation. The success of these systems undoubtedly will ensure that the 286 will be around for a long time.

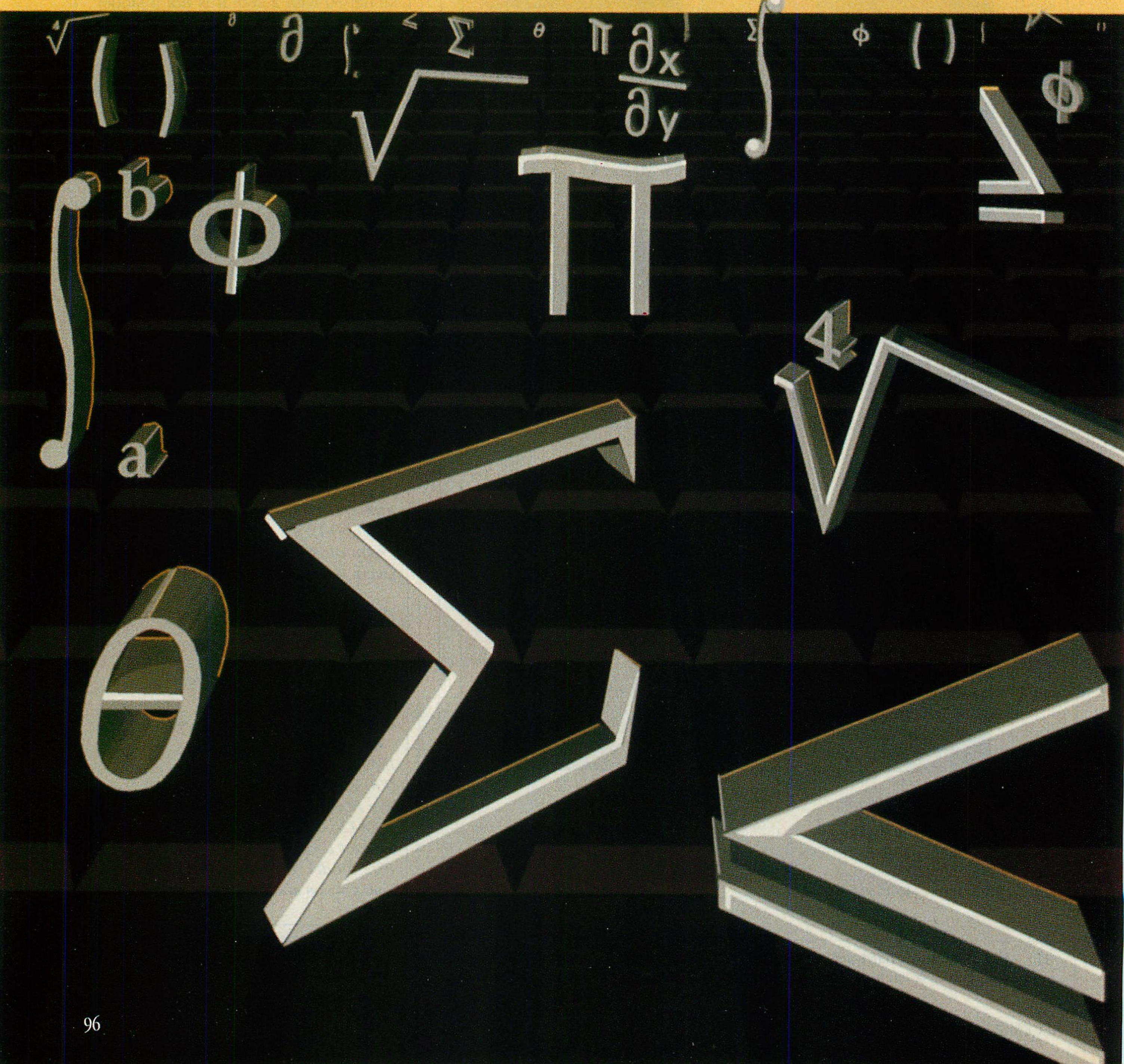
AST Research, Inc.
2121 Alton Avenue
Irvine, CA 92714
714/863-1333
Premium Workstation/286
CIRCLE 340 ON READER SERVICE CARD

Dell Computer Corporation
9505 Arboretum Blvd.
Austin, Texas 78759-7299
800/426-5150; 512/338-4400
System 220
CIRCLE 341 ON READER SERVICE CARD

IBM Corporation
900 King Street
Rye Brook, NY 10573
800/426-2468; 800/447-4700 for nearest dealer
PS/2 Model 50Z; PS/2 Model 30 286
CIRCLE 342 ON READER SERVICE CARD

David Claiborne is a technical manager for JAYCOR in Edgewood, Maryland, and a contributing editor to PC Tech Journal.

FORTRAN



Meets OS/2

What happens when a 30-year-old programming language meets a 1-year-old operating system? FORTRAN compilers from IBM and Microsoft make the transition from DOS to OS/2.

JOHN VOGLEWEDE

The patriarch of high-level programming languages is one of the first assimilated into the OS/2 family. Thus far, two DOS FORTRAN compilers have been revised to run under OS/2: IBM FORTRAN/2, developed by Ryan-McFarland (RM), which was acquired by Austec Inc. in mid-1987, and Microsoft (MS) FORTRAN 4.1. Both require OS/2 and an 80286 machine. (DOS versions of these compilers were reviewed in "FORTRAN Perspectives," John Voglewede, June 1987, p. 92.) Austec plans to release its own compiler by December 1988.

FORTRAN owes its popularity in part to the well-developed, widely observed FORTRAN-77 (known formally as ANSI X3.9-1978). This industry-wide standard has been adopted by the U.S. Government, endorsed by the U.S. Department of Defense, and recognized by the International Standards Organization (ISO). The standard and the maturity of the language ensure program portability that is unmatched among mainframe and microcomputer languages. Applications developed on computers of any size move easily between systems. Both FORTRAN compilers conform to the full standard under OS/2, with no significant omissions.

Where does FORTRAN fit into the OS/2 picture? FORTRAN under OS/2 will attract the same scientific- and engineering-oriented problem solvers who are currently using FORTRAN under DOS or on a minicomputer or mainframe. OS/2's additional memory will make it possible to use large FORTRAN applications on the PC.

Unfortunately, FORTRAN is unable to exploit many other OS/2 features. The OS/2 API is best suited for languages such as C and Pascal, which have robust data-structure capabilities. Modern PC applications emphasize the user interface, and FORTRAN does not help the developer with this part of the application—OS/2 Presentation Manager is not supported. In addition, FORTRAN is not recursive—that is, a FORTRAN subprogram cannot reference itself, and some library routines are not reentrant. This restricts the use of OS/2 threads in a FORTRAN environment.

The increase in RAM under OS/2 allows the creation of the large arrays that characterize many FORTRAN applications. Intel 286- and 386-class machines provide as much as 16MB of physical memory in OS/2 —a refreshing departure from the DOS 640KB limit. OS/2 also provides a virtual memory environment: a program can define more data than there is RAM, and the operating system swaps segments into memory as needed.

Despite OS/2's additional memory, developers still must contend with limitations imposed by the 80286. The compilers provide compilation options that trade off between maximum available memory and execution speed. Microsoft provides three memory models—medium, large, and huge; IBM provides only large and huge.

In the medium memory model, the code size can be larger than 64KB, but all data must fit into a single 64KB segment. The large model allows more than 64KB of data, but no formal array arguments passed to a function or subroutine can be larger than 64KB. The huge model allows array arguments of any size.

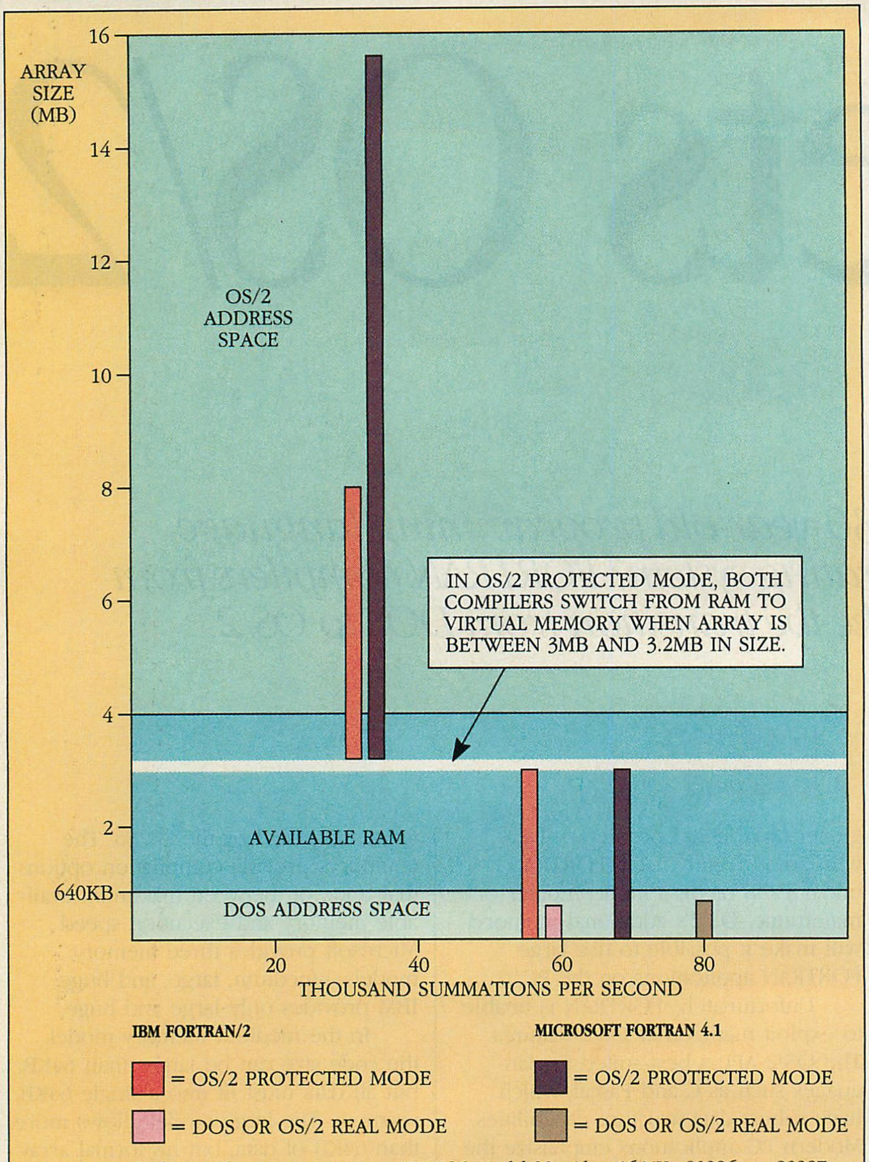
The medium model produces the fastest execution speed: since the data segment is known, all data can be referenced by a 16-bit offset. The huge model is the slowest: no simplifying assumptions can be made, so worst-case code is generated. The default for both compilers is the large model.

CAN YOU TEACH AN OLD DOG . . . ?

In most respects, compiling, linking, and executing programs using these compilers under OS/2 is similar to doing so under DOS. Both compilers run under OS/2 protected mode, OS/2 real mode (the DOS compatibility box), or DOS. The compilers can generate executable files that run only under OS/2, only under DOS, or in

COMPUTER GRAPHIC • TONEY SIDWELL

FIGURE 1: ***FORTRAN Performance and Array Size***



Both compilers were run on an IBM PS/2 Model 80 with a 16-MHz 80386, an 80387, 4MB RAM, and a 70MB hard disk. When tested under OS/2, each program was the only executing process.

The time needed to sum a single element in an array is a function of array size. Under the DOS 640KB memory limit, Microsoft's compiler outperforms IBM's; arrays are limited to a maximum of 400KB. Under OS/2 protected mode, both can have very large arrays (Microsoft to 15.6MB, IBM to 8MB), but performance slows considerably when they switch from RAM to virtual memory.

both environments by compiling and linking the program for protected mode and *binding* it (see "Family Ties," David Schmitt, June 1988, p. 124). Binding resolves references to the OS/2 dynamic link libraries (DLLs) so the program runs in either mode (see "OS/2's Dynamic Link," Mary DeWolf and Ted Mirecki, September 1988, p. 100). MS FORTRAN provides a BIND utility to generate such programs; FORTRAN/2 provides no such utility. The process is explained but is not provided with FORTRAN/2—the package refers the user to the BIND utility in the IBM OS/2 Programmer's Toolkit.

Beyond some obvious and some subtle differences, the packages deliver similar capabilities and have like requirements. MS FORTRAN runs under virtually any version of DOS, but FORTRAN/2 requires DOS 3.3 or later; both can generate 286-specific code. Each package is missing a few features from its repertoire. Table 1 summarizes system requirements for the compilers.

FORTRAN/2 does not provide a single-step, compile-and-link command, nor does it provide for multiple source files or a wild-card naming convention in the command line. The INCLUDE directive allows inclusion of multiple source files, but nested INCLUDE statements are not permitted; neither does the compiler generate code for checking the validity of array subscripts during runtime. The FORTRAN/2 documentation describes interfacing with assembly language programs, but not with other languages.

Unlike MS FORTRAN, FORTRAN/2 lacks a source-code editor and an alternate math package. MS FORTRAN includes two emulation packages for systems that do not have a math coprocessor. The emulator library offers strict compatibility with the coprocessor's calculations; the alternate math library provides faster calculations at some expense in accuracy.

Table 2 summarizes the language features of each compiler. Because each conforms to the full ANSI FORTRAN-77 standard, their overall language features are similar. The ANSI FORTRAN-66 entry in the table refers to the older standard, which differs semantically from FORTRAN-77.

Table 2 also lists compiler extensions to FORTRAN-77. Free-format source code (which is not supported by FORTRAN/2) is a useful feature that recognizes uppercase letters, digits, and 13 special characters. MS FORTRAN recognizes all ASCII characters; of the special characters, FORTRAN/2 recognizes only those listed in the table.

The IMPLICIT NONE statement requires all variables to be explicitly declared. This construct, not included in the FORTRAN-77 standard, is now featured by both packages. FORTRAN/2 offers it as a language construct, MS FORTRAN as a compiler option that can be invoked from the command line. The latter method is more convenient because IMPLICIT NONE is a debugging aid used to identify variables not explicitly declared. Microsoft's approach lets the developer use the feature without disturbing the source code of the program being debugged.

The NAMELIST construct is missing from both compilers. Common among mainframe compilers, NAMELIST names a group of variables that can be easily referred to in I/O statements.

Data typing is a standard feature in FORTRAN-77, but data-type lengths are not—that is, REAL is a recognized statement but REAL*4 is not. FORTRAN-77 requires a common but unspecified length for both REAL and INTEGER data types. If the default length of a

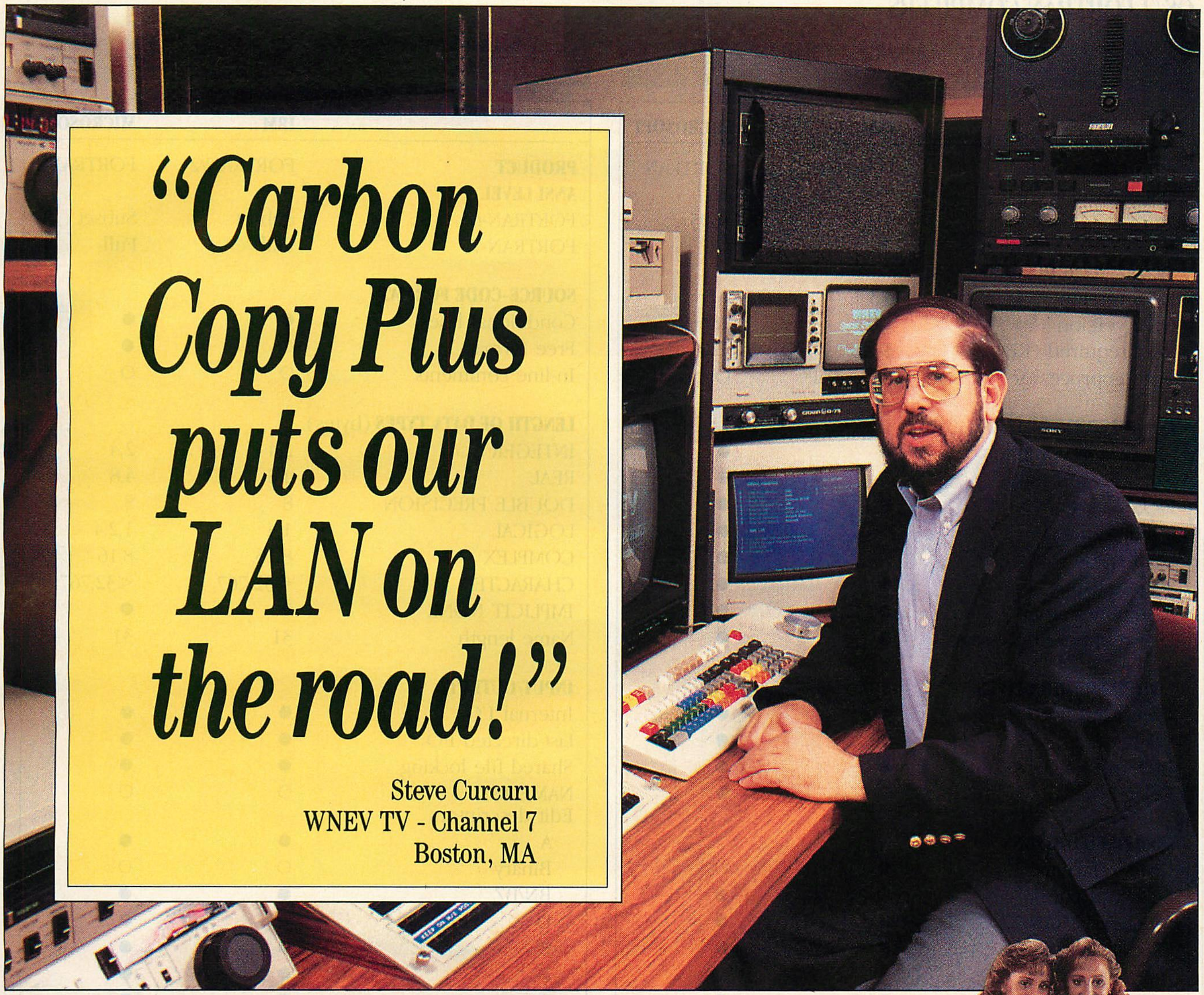

"Carbon Copy Plus puts our LAN on the road!"
Steve Curcuru
WNEV TV - Channel 7
Boston, MA

TABLE 1: *Compiler Specifications*

	IBM	MICROSOFT
PRODUCT	FORTRAN/2	FORTRAN
VERSION	1.0	4.1
PRICE	\$595	\$695
SYSTEM REQUIREMENTS		
DOS version	3.3	2.0+
OS/2 version	1.0+	1.0+
RAM required (KB)	384	320
Math coprocessor	Optional[a]	Optional[a]
COMPILER INVOCATION		
Single-step compile	●	●
Compile and link	○	●
DOS path names	●	●
Multiple files	○	●
Wild cards	○	●
Source listing	●	●
Assembly output	●	●
Line-numbered message	●	●
COMPILER DIRECTIVES		
Command line	●	●
Source file	●	●
Include	●	●
Nested	○	●
COMPILER OPTIONS		
Cross-reference	●	○
Subscript check	○	●
Strict FORTRAN-77	●	●
Global save	●	●
Interface		
Assembly	●	●
BASIC	○	●
C	○	●
Pascal	○	●
80286-specific code	Optional[b]	Optional[b]
LINKER OPTIONS		
Memory overlays	●	●
Preset data values	○	○
LIBRARY OPTIONS		
8087/80287	●	●
8087/80287 emulation	●	●
Alternate math	○	●
OTHER PROGRAMS		
Linker	●	●
Debugger	●	●
Source-code editor	○	●
Librarian	●	●

● = *Yes* ○ = *No*

[a] *80x87 will be used if present.*
[b] *Developers can choose to generate 80286/287-specific code to run under DOS or OS/2, or code that will run under both.*

The packages are similar, but FORTRAN/2 requires more memory and a stricter DOS interpretation. FORTRAN/2 does not support compile-and-link, wild cards, or multiple files, nor does it interface to other high-level languages.

TABLE 2: *Language Features*

	IBM	MICROSOFT
PRODUCT	FORTRAN/2	FORTRAN
ANSI LEVEL		
FORTRAN-66	Subset	Subset
FORTRAN-77	Full	Full
SOURCE-CODE FORMAT		
Conditional lines	●	●
Free format	○	●
In-line comments	○	○
LENGTH OF DATA TYPES (bytes)		
INTEGER	2,4	2,4
REAL	4,8	4,8
DOUBLE PRECISION	8	8
LOGICAL	1,4	1,2,4
COMPLEX	8,16	8,16
CHARACTER	<32,767	<32,767
IMPLICIT NONE	●	●
Name length	31	31
INPUT/OUTPUT		
Internal I/O	●	●
List-directed I/O	●	●
Shared file locking	●	●
NAMELIST	○	○
Edit descriptors		
A	●	●
Binary	○	○
BN/BZ	●	●
D	●	●
E	●	●
F	●	●
G	●	●
Hollerith	●	●
I	●	●
L	●	●
Octal	○	○
P	●	●
S/SP/SS	●	●
T/TL/TR	●	●
X	●	●
Z (Hexadecimal)	○	●
CHARACTER SET		
Lowercase	●	●
Nonstandard characters	●[a]	●
NONSTANDARD FUNCTIONS		
Bit manipulation	●	●
COMPLEX*16 operators	●	●
Date/time	●	●
OS/2 system calls	●	●
RAM operators	●	●
String operators	●	●

● = *Yes* ○ = *No* [a] *Only <, >, _, and \\.*

Both compilers support ANSI FORTRAN-77 and many extensions. Microsoft's compiler boasts a few features that are missing from the IBM package, including free format, hexadecimal edit descriptors, and a full ASCII character set.

REAL variable is four bytes, the default length of an INTEGER variable must also be four bytes. The available, albeit nonstandard, data-type lengths are also shown in table 2.

The documentation for both compilers is lucid and comprehensive. Both vendors attempt to identify extensions to the ANSI FORTRAN-77 standard; Microsoft's documentation sets them apart with a different color type. Both offer strict FORTRAN-77 language interpretation as a compiler option. IBM's documentation stresses portability (the transfer of source code), program conversion, and language extensions, emphasizing the importance of writing programs to run on more than one computer system. The license agreement for each is quite liberal.

The FORTRAN/2 agreement permits the redistribution of library modules when the application depends on them. Distribution diskettes must be labeled with the IBM copyright notice, but no explicit permission or royalty payments are needed. Similarly, Microsoft's license grants a royalty-free right to reproduce and distribute .EXE files created with the compiler. How the library information is used influences the copyright requirements. Users considering commercial applications based on either compiler must read the fine print of the applicable agreement.

THE WORLD ACCORDING TO OS/2

For systems configured with less than the 16MB of RAM available under OS/2, the OS/2 *storage overcommitment* feature permits compiling, linking, and executing FORTRAN programs with memory requirements far exceeding the memory available. This is accomplished by discarding memory segments that are no longer required, swapping segments for those currently required, and combining segments.

In theory, storage overcommitment should be limited only by disk storage space; in practice, other limits usually are reached first. In FORTRAN/2, for example, maximum program size (including code, data, and library support) is 16MB. Even this declared limit is optimistic in practice.

Figure 1 shows the actual availability of large amounts of RAM through each compiler. To generate the information for this figure, a simple FORTRAN program was written with a large one-dimensional array of REAL numbers. A DO-loop gives each element an initial value. A second DO-loop sums all elements, and the program calculates the time needed to perform the summation. The figure shows the average time to sum each element in the array (the rate at which elements can be summed) as a function of the size of the array.

The DOS 640KB limit dictates that the maximum size of an array must be smaller than this limit. For this review, approximately 500KB was available for a FORTRAN array under DOS. Program overhead and DOS itself consume the remaining space. The two short vertical bars completely contained within the 640KB DOS address space represent each of the two programs running under DOS. (Code run under DOS was slightly faster than the OS/2 counterpart; also, MS FORTRAN was slightly faster than IBM FORTRAN/2.)

Both FORTRAN packages compiled, linked, and executed the benchmark programs very quickly, with a few notable exceptions.

Under OS/2, the array size was increased well beyond the DOS 640KB limitation. As the array size approached the 4MB physical memory limit of the test machine, OS/2's virtual memory management came into play. Logical segments of the program that cannot be placed in RAM are temporarily stored on hard disk and transferred when required. Considerable disk swapping is associated with this operation; as a result, the average access time per element of the array more than doubled.

Because swapping to disk is expensive, developers should write code that avoids successive references to different segments. For example, this code fragment initializes an array:

```
      :
      INTEGER TAB(16384,10)
      INTEGER I, J
      DO 100 I = 1, 16384
         DO 100 J = 1, 10
            TAB(I,J) = 0
100   CONTINUE
      :
```

FORTRAN stores arrays in column-major order so that all values for a single column in the array are adjacent in memory. In the example above, the array TAB has 10 columns; each column requires a 64KB memory segment (16,384 rows per column, 4 bytes per integer). Each iteration of the inner loop on J requires a different memory segment from the previous iteration. In the worst case (only 1 of the 10 segments can be kept in memory), OS/2 will need to perform 163,840 segment swaps between disk and memory to initialize TAB.

The solution for this example is to swap the two DO loops, so that an entire column is initialized at a time. Only 10 segment swaps are needed to initialize TAB in the worst case. This example shows that although OS/2 provides virtual memory, it is still important for the developer to understand the nuances of memory management.

Using virtual memory imposes practical limitations. Compiler and linker limits, such as the number of logical segments per program, must increase to handle large programs. These steps were taken with both compilers. The documentation for FORTRAN/2 states that a program can take as much as 16MB, but a considerably lower limit was found to be the case. (IBM offered no explanation for the discrepancy.) The IBM linker succumbed to an array of slightly more than 2 million elements (8 million bytes) with the declaration that the object module was invalid.

In contrast, MS FORTRAN compiled, linked, and executed programs with arrays of more than 3.9 million elements (15.6 million bytes) before providing the fatal-error message "too many groups, segments, and class names in one module." In both cases, far more memory is available to the user than under DOS.

COMPUTATIONAL HERITAGE

IBM FORTRAN/2 makes a strong showing in the OS/2 environment. It supports extensive compiler optimization, error diagnostics, and interactive debugging facilities. The IBM compiler was written by Ryan-McFarland (Austec) and is being sold under the IBM label.

Substantial improvements were made between IBM Professional FORTRAN 1.0 (also written by Ryan-McFarland) and FORTRAN/2. New features include the interactive debugger, math coprocessor emulation, and a COMPLEX*16 data type—these features were covered in the review of IBM FORTRAN/2's most recent predecessor, RM FORTRAN 4.1 (see the aforementioned June 1987 review).

In addition to OS/2 compatibility, several new compiler options and envi-

ronment variables are provided to promote easier operation. Compiler and linker limits now accommodate larger system configurations. The product continues to emphasize mainframe compatibility with the addition of several enhancements in this area; for example, all textual messages are isolated to facilitate language translation. The FORTRAN/2 compiler is certainly a top choice for ease of use.

The manual contains comparisons between FORTRAN/2 and several mainframe FORTRAN compilers. Popular extensions to the FORTRAN-77 standard are included to facilitate program transfer from other hardware. FORTRAN/2 is also the serious candidate for users interested in portability.

Installation of FORTRAN/2 is easy with the automatic installation procedures. Proper operation of the compiler under OS/2 real or protected mode is quick and easy once the configuration files (AUTOEXEC.BAT and OS2INIT.CMD) are established. The product provides overlay support for DOS, multitasking under OS/2, and file sharing over networks. Programs can make calls to the OS/2 application program interface (API). The documentation carefully explains the details of each operation.

An interactive debugger, the Interactive Symbolic Debug program, is an added feature. The debugger allows the programmer to view the source code, set breakpoints, monitor program execution, and examine, modify, and trace changing variables. The debugger works under DOS and both real and protected modes of OS/2.

FOR THE PROFESSIONAL

MS FORTRAN comes with massive materials: three binders of technical information, several pamphlets and guidebooks, two keyboard templates, and 10 diskettes. Microsoft also offers the information on 3.5-inch media at no charge. The company is holding the line on the formidable RAM requirement for this product (320KB required and 512KB recommended).

One significant change is an improved version of the Microsoft source-code and assembly-level debugger, CodeView (reviewed in "Multilevel Debugger," Mark S. Ackerman, March 1987, p. 90). This windows-oriented utility lets users track down logical errors in programs. Like its IBM counterpart, CodeView displays source code or assembly language code, indicates which line is about to be executed, and dynamically watches the values of variables. CodeView also switches screens to display program output and perform many other related functions.

One of CodeView's new features is to perform multilanguage expression evaluation. Its built-in language interpreter evaluates C, BASIC, FORTRAN, or Pascal expressions. It supports overlays, the 8087, the 80286, and expanded memory. Bound programs cannot be debugged using CodeView; however, programmers can use the OS/2 version of CodeView to debug a protected-mode version of the program.

MS FORTRAN is not for the casual user. Its three memory models (medium, large, and huge), three floating-point math libraries (8087, 8087 emulator, and alternate math), and two methods of using floating-point instructions (in-line instructions or calls to the floating-point library routines) offer many choices and require careful planning, because not all combinations of these options are meaningful. The automatic installation procedures included on the setup diskette make loading the compiler on a hard disk easy. The user need only answer questions on compiler configuration options concerning subjects such as compatibility with earlier MS FORTRAN compilers and the memory model or math package being used. A help tutorial is available.

MS FORTRAN is not significantly changed from its previous incarnation, except for OS/2 support. It still lacks a cross-reference option, and compiler directives are case-sensitive, requiring upper- and lowercase directives to run. In addition, the compiler command line is annoying to the veteran; using DOS .BAT or OS/2 .CMD batch files helps. Microsoft also permits frequently used options to be placed as environment variables in the operating system.

One important addition to MS FORTRAN is the Microsoft Editor, a development tool that runs under OS/2 or DOS 2.1 and later. FORTRAN/2 has no such feature. The MS full-screen editor allows the user to create source files, customize editing functions, and invoke compilers (or other utilities such as assemblers).

The Editor's ability to handle complex files originates with its use of windows. The screen splits into as many as eight areas, each displaying a different part of the same file or multiple source files. The Editor can be used as a simple text editor or more effectively to write, compile, and link programs from within the editor. If the compile fails, the user can view the errors, rewrite the program, and recompile, all with-

TABLE 3: *Benchmark Results*

MODE	IBM FORTRAN/2 DOS	IBM FORTRAN/2 OS/2	MICROSOFT FORTRAN DOS	MICROSOFT FORTRAN OS/2
MINIMUM (1 line)				
Compiled size	346	346	250	250
Linked size	9,197	1,049	2,365	2,817
Compile time	00:02	00:02	00:05	00:05
Link time	00:03	00:01	00:01	00:02
Runtime	00:01	00:01	00:01	00:01
SYNTH1 (27 lines)				
Compiled size	1,173	1,173	1,226	1,226
Linked size	29,904	1,190	22,904	25,822
Compile time	00:03	00:03	00:06	00:06
Link time	00:07	00:01	00:03	00:05
Runtime	00:01	00:01	00:01	00:01
SYNTH2 (207 lines)				
Compiled size	5,758	5,758	7,827	7,827
Linked size	33,200	4,470	25,608	32,478
Compile time	00:11	00:11	00:13	00:14
Link time	00:07	00:02	00:03	00:07
Runtime	00:01	00:01	00:01	00:01
SYNTH3 (2,007 lines)				
Compiled size	—[a]	—[a]	86,043	—[b]
Linked size	—	—	56,616	—
Compile time	—	—	01:28	—
Link time	—	—	00:06	—
Runtime	—	—	00:01	—
ITERATE (23 lines)				
Compiled size	947	947	853	853
Linked size	29,728	1,134	22,776	5,310
Compile time	00:02	00:03	00:06	00:06
Link time	00:06	00:01	00:03	00:05
Runtime	01:24	01:25	01:04	01:14
BIGARRAY (22 lines)				
Compiled size	161,940	161,940	862	862
Linked size	189,696	161,585	182,744	184,998
Compile time	00:47	00:48	00:05	00:05
Link time	00:15	00:12	00:07	00:12
Runtime	00:01	—[c]	00:01	00:01[d]
BIGGER (22 lines)				
Compiled size	2,579,384	2,579,384	2,478	2,478
Linked size	2,589,696	2,562,097	2,582,744	2,585,766
Compile time	12:13	12:42	00:05	00:06
Link time	05:25	04:39	03:41	03:37
Runtime	—[e]	—[c]	—[e]	00:01[d]
BIGERROR (22 lines)				
Errors detected				
Compilation	12	12	12	12
Execution	0	0	1	1
Passes required	5	5	5	5
Errors undetected	2	2	1	1
Line numbers	14, 15	14, 15	14	14

All sizes are in bytes; all times are in seconds.

All benchmarks were run on a 16-MHz IBM PS/2 Model 80 with an 80387.

The tests were run using Microsoft OS/2 1.0 and DOS 3.3.

[a] *Available compiler workspace exceeded.*
[b] *Protection violation.*
[c] *Executes with incorrect results.*
[d] *Microsoft OS/2 1.1 produces correct results.*
[e] *Insufficient memory for execution.*

MS FORTRAN is the faster in an overall close competition; however, it is worth noting that the benefit of OS/2 is not higher speed, but larger memory. Both compilers struggled unsuccessfully with the large SYNTH3 program under OS/2. Only MS FORTRAN could execute BIGARRAY and BIGGER correctly under OS/2 1.1; FORTRAN/2 took forever to compile BIGGER.

out leaving the editor. It can be customized, allowing the user to reassign editing functions to different keys and specify the function assignments in a boot file. The Editor automatically recognizes these assignments each time it is loaded. A user can make up key assignments to match a favorite word processor and can construct macros.

SAME AS IT EVER WAS

With one addition, the benchmarks applied in this review are identical to those applied in the June 1987 review. These benchmarks include MINIMUM (the smallest allowable FORTRAN program); SYNTH1, SYNTH2, and SYNTH3 (27-, 207-, and 2,007-line programs, respectively); ITERATE (a program with 1 million arithmetic operations in a nested DO loop); BIGARRAY (a program with a 200 * 200, or 40,000-element array); and BIGERROR (a replicate of BIGARRAY with several errors included). The benchmarks were produced with the extension .F77 in the previous article and are available for downloading on PCTECHline.

The new benchmark BIGGER.FOR is the only one specifically tailored for testing under OS/2. It is similar to BIGARRAY except that it uses an array increased to 640,000 (800 * 800) elements, or 2.56 million bytes at 4 bytes per element. A single DATA statement similar to that used in BIGARRAY sets the initial conditions. BIGGER adds up all elements and displays the result. The source code is still only 22 lines, and the program appears quite small; however, the resulting 2.56MB array is significantly larger than the PC's 640KB limit under DOS.

The test system used to compile, link, and execute the benchmarks was an IBM PS/2 Model 80 with 16-MHz 80386, an 80387 math coprocessor, 4MB of RAM, and a 70MB hard disk. All operations were timed with the operating system TIME function. Table 3 lists the results under DOS and OS/2.

Both compilers produce .EXE files that detect a math coprocessor and use it if present. Other options, such as MS FORTRAN's three memory models, three floating-point math libraries, and variations for including floating-point instructions in a program, can alter the compile, link, and execution times. Where compilation or linking features were offered, the standard or default settings were used.

The program sizes reported in table 3 are actual for the .OBJ or .EXE file generated, regardless of whether .DLLs or other modules were required

to load and execute the file. For example, the .EXE program size for the DOS version of SYNTH1.EXE is 29,904 and 22,904 bytes (for FORTRAN/2 and MS FORTRAN, respectively). Either file loads and executes under DOS with no additional files (beyond DOS itself).

The .EXE program size for the OS/2 version of this program is 1,190 and 25,822 bytes (for FORTRAN/2 and MS FORTRAN, respectively). The FORTRAN/2 compiler produces extremely compact code, but it requires a DLL—FORTRAN.DLL (77,728 bytes)—to run. The Microsoft file loads and executes in an OS/2 environment with no additional files (beyond OS/2).

Most of the benchmark programs compiled, linked, and executed very quickly, as expected. One exception is the 12 to 13 *minutes* FORTRAN/2 needed to compile BIGGER, compared with the 5 to 6 *seconds* MS FORTRAN took. The object file size for BIGGER and an examination of the .EXE file with a debugger show that data initialization is deferred from compilation to link time in the Microsoft compiler. In contrast, the size of the IBM-compiled program shows that the initial data values are generated during the compile process itself.

The compilation speed of IBM FORTRAN/2 is considerably improved over the previous version. In BIGARRAY and BIGGER, the compiler continues to struggle with the DATA statement, which could be replaced with a DO-loop or other language construct. In the June 1987 review, the product took more than 28 minutes to compile BIGARRAY on a 4.77-MHz 8088 machine with an 8087.

Improvements to the compiler and use of a 16-MHz 386 machine with a 387 brought compilation time down to 48 seconds; however, BIGGER took considerably longer—more than 12 minutes—on the newer machine. Preliminary work with the Austec compiler, which is scheduled for December 1988 release, showed compile times 1.5 to 2 times faster for all benchmarks than those that are reported here for FORTRAN/2. Austec has sent the fix to IBM, but IBM has not indicated when it will be added to FORTRAN/2.

IBM's compiler completed only the DOS version of BIGARRAY successfully. Under OS/2, the program compiled and ran without obvious error, but produced incorrect results.

Neither of these compilers successfully completed BIGGER, which is understandable under DOS, where insufficient memory is available to run the problem. Under OS/2 1.0, the program compiled, linked, and executed without errors and then produced incorrect results—with no warning. Microsoft acknowledges the problem as a known OS/2 1.0 error: if the OS/2 loader is used to initialize an array whose size is greater than 64KB, any segment that is entirely filled will contain zeros. Because the problem is due to an OS/2 loader error, any OS/2 FORTRAN compiler (or any language compiler using the loader to initialize arrays larger than 64KB) will fail similarly.

Running BIGGER under a beta version of Microsoft OS/2 1.1 solves the problem for MS FORTRAN, but not for FORTRAN/2. According to IBM, BIGGER can be compiled, linked, and successfully executed under IBM OS/2 1.1. In both compilers, replacing the DATA statement with a DO-loop allows correct execution of BIGGER. The failure with no warning under OS/2 1.0 is a serious matter.

Both compilers had trouble with SYNTH3. MS FORTRAN handled this benchmark successfully in DOS mode,

but not under OS/2 where the compiler's only complaint was a warning message, "function too large for post-optimizer." Microsoft traced the difficulty to a problem in the second pass of the compiler and is working on a correction. FORTRAN/2 could not handle the problem in either DOS or OS/2 mode. The problem under DOS is the compiler's limit of 1,800 to 1,900 lines of code, according to an Austec representative; no explanation was offered by Austec or IBM for the failure to compile SYNTH3 under OS/2.

Each compiler did rather well with BIGERROR (except the situation where the uninitialized variable SUM appears for the first time on both the right and left sides of an expression). Twelve different errors were detected during five compilation passes, and MS FORTRAN detected one more error during execution with its subscript-checking capability. Also interesting is an early error detected by FORTRAN/2, which reported that using lowercase characters in a comment was not FORTRAN-77 standard. ANSI X3.9-1978 actually allows *any* character (in a comment field) that can be represented. The benchmark programs were converted to uppercase to eliminate this problem.

Examining simultaneous compilation and execution in an OS/2 multiprocessing environment revealed great variability in the time required to complete the tasks. Similar results were reported previously (see "Enter OS/2," Ted Mirecki, November 1987, p. 52). In OS/2 mode, performance depends not only on the number and type of independent actions or threads being run, but also on the system conditions (that is, the MAXWAIT, PRIORITY, and TIMESLICE parameters of OS/2). When two identical processes are run simultaneously, a more than doubling of the execution time can be expected. This does not prevent the simultaneous compilation, linkage, and execution of various programs under OS/2 FORTRAN. The processes need not be related to FORTRAN.

THE BOTTOM LINE

Neither of these OS/2 FORTRAN compilers is outstanding, although both have improved since *PC Tech Journal* reviewed the DOS-only versions. Each package continues to deliver reasonable performance under DOS; however, they have yet to follow through with the same level of consistency under OS/2 as evidenced by the problems with the SYNTH3 benchmark even under OS/2 1.1. With caution, the advantages of mainframe FORTRAN are available; but the apparent immaturity of these products suggests that although they can be used under DOS, they may be better left to ripen before being used under OS/2.

IBM Corporation
P.O. Box 1328-W
Boca Raton, FL 33429
800/426-2468; 800/447-4700 for nearest dealer
FORTRAN/2

CIRCLE 335 ON READER SERVICE CARD

Microsoft Corporation
16011 N.E. 36th Way
P.O. Box 97017
Redmond, WA 98073
800/426-9400; 206/882-8080
FORTRAN 4.1

CIRCLE 336 ON READER SERVICE CARD

John Voglewede works for the U.S. Nuclear Regulatory Commission. He holds degrees in physics, computer science, and mechanical engineering. His previous article on FORTRAN compilers appeared in June 1987.

PRICED TO SAVE, SHIPPED FAST!!!

FRONTRUNNER *(a new Ashton Tate product created by Apex Software)*

FrontRunner generates memory-resident **pop-up** applications using the dBASE III Plus or dBASE IV programming language. Once your memory-resident program is created, **pop-up** your program while using Lotus 1-2-3, any dBASE program, Clipper, Basic, any word processor, DOS or any program written in any language. Pop-In and Pop-Out of any software package instantly. All data files and indexes created are dBASE compatible. FrontRunner applications can be compiled into binary files for protected distribution and unparalleled speed. The user selects a hot-key to call up an application. Use the unique POWERKEY feature to define additional hot-keys within the application. A powerful paste command allows the user to extract data from a FrontRunner application into a word processor, spreadsheet or other application. Use FrontRunner for **pop-up** help screens, account listings, phone directories, schedules, random notes, printing mailing labels.... anything imaginable.
FrontRunner....List: **$295**, Ours: **$195** *(new low price!)*

GREENLEAF LIBRARIES

Functions

C source, assembler source, and binary libraries of 225 functions for many compilers. Emphasizes tight functional groupings to minimize loading code which your application may never use. Manual's 250 pages help select functions, as do demos, bulletin board.

Communications

Communicate from within your own C programs! Over 120 functions and demo programs in C and assembler source to set up interrupt-driven asynch communications for up to 16 channels. Up to 9600 baud, ASCII or binary, any parity or word length, 8250 UARTs, Xon/Xoff and Xmodem, WideTrack receive. Goodbye separate communications software. Specify compiler.

Data Windows

Windows, menus and data entry do work together...when you utilize Greenleaf's screen architect. This smooth screen designer offers device independence, logical windows, table driven data entry and economical pricing. Source code is also available.

	List	Ours
Communications	**$185**	**$139**
Functions	**$185**	**$139**
Data Windows	**$295**	**$249**
Complete 3 in 1 Pack	**$665**	**$475**

Shopping List for the Power Workbench

	LIST	US
ASSEMBLER		
Microsoft Macro Assembler with Utilities	150	109
PASM 86 *by Phoenix, Macro Assembler*	195	109
ASSEMBLER Support		
Btrieve Softcraft's File Manager	245	175
GSS CGI....*Device independent graphics*	495	425
BASIC		
Microsoft BASIC Interpreter..*for XENIX*	350	249
Microsoft QuickBASIC...*Ver 4.0*	99	66
Turbo BASIC.......*NEW from Borland*	100	75
BASIC LIBRARIES & UTILITIES		
Btrieve Softcraft's File Manager	245	175
GSS CGI....*Device Independent Library*	495	425
Halo Graphics by Media Cybernetics	325	249
C LANGUAGE COMPILERS		
C86 PLUS by Computer Innovations	497	397
Lattice C Compiler Now ver 3.2	450	299
Let's C Compiler from Mark Williams Co	75	55
Mark Williams C full development system	495	369
Microsoft C Compiler with free CODEVIEW	450	295
Microsoft QuickC....*Special Price*	99	66
Turbo C.......*New from Borland*	100	75
C LIBRARIES—Communications		
Asynch Manager by Blaise	175	135
Greenleaf Communications	185	139
Essential Communications	185	125
Essential Communications Plus	250	189
C LIBRARIES—FILE MANAGEMENT		
Btrieve Softcraft's File Manager	245	175
Btrieve/N File Management for Networks	595	445
Ctree by Faircom,with full source	395	299
Rtree...*Report Gen. for Ctree*	295	235
Ctree & Rtree.....*Special Combination*	650	499
dBC ISAM Accesses dBase files	250	175
with Source code	500	349
dBC III Plus multiuser	750	595
with Source code	1500	1195
Opt Tech Sort Super fast sort for Btrieve	149	105
XQL....SQL *from NOVELL*	795	595
C LIBRARIES—Graphics		
Essential Graphics...*no royalties*	299	225
GSS CGI....*Device independent graphics*	495	425
GSS Metafile Interpreter stores images	295	265
Halo '88 by Media Cybernetics	325	249
Halo for Microsoft Languages	595	434
C LIBRARIES—Screen Design		
Curses from Lattice,UNIX lookalike	125	99
with source	250	199
C Worthy....*by Custom Design Systems*	195	159
C Worthy with Forms	295	269
Greenleaf Data Windows	295	225
Microsoft Windows Dev. Toolkit	500	365
Panel Plus by Roundhill	495	395
View Manager for C,Blaise	275	199
Vitamin C...*Creative Programming*	225	198
VC Screen...*Source code Generator*	100	81
Windows for C	195	149
Windows for Data	295	259
Zview....*Data Management Consultants*	245	175
C UTILITY LIBRARIES		
Basic C....*Basic-like routines for C*	175	139
Blaise C Tools Plus/5.0....*f/MsC & QuickC*	129	99
Blaise Turbo C Tools...*f/TurboC*	129	99

	LIST	US
C Food Smorgasbord by Lattice	150	109
C Utility Library by Essential,300 functions	185	119
Greenleaf Functions	185	139
PforCe by Phoenix, vast library	395	199
OTHER TOOLS		
BASTOC...*JMI,Translates BASIC to C*	495	399
dBX Translator......*dBASE to C translator*	550	419
with Library Source	950	725
Pre/C....*by Phoenix, like UNIX lint*	295	154
PC-LINT..*by Gimpel, subset of UNIX Lint*	139	125
COBOL		
Micro Focus COBOL/2	900	795
Micro Focus Toolset	900	795
Micro Focus Personal COBOL	149	134
Microsoft COBOL inc. COBOL Tools	700	499
for XENIX	995	749
RM/COBOL...*by Ryan McFarland*	950	675
RM/COBOL 85...*ANSI 85*	1250	895
COBOL Support		
Btrieve Softcraft's File Manager	245	175
GSS CGI....*Device independent graphics*	495	425
Halo......*from Media Cybernetics*	325	249
RM/Screens..*Screen generator*	395	335
RM/Net+ 5...*R/M COBOL networking*	300	249
DBASE & RELATED PRODUCTS		
Applications Plus....*Fox & Geller*	299	249
Brief & dBrief..*Editor/Macro lang for DBase*	275	219
Clipper.....*Nantucket's DBase Compiler*	695	399
DATA-p..........*Wallsoft*	60	50
dBC III Plus supports multiuser commands	750	595
with Source code	1500	1195
dBC ISAM...*accesses dBase files*	250	175
with Source code	500	349
dBX Translator......*dBASE to C translator*	550	419
with Library Source	950	725
dFlow.......*Wallsoft*	149	124
Documenter.......*Wallsoft*	295	244
FoxBase+.....*Fox Software*	395	219
Multiuser version	595	349
QuickCode Plus........*Fox & Geller*	295	170
QuickEntry...........*Fox & Geller*	99	59
QuickReport...........*Fox & Geller*	295	170
UI Programmer....*Wallsoft*	295	244
FORTRAN Compilers & Utilities		
Btrieve: Softcraft's File Manager	245	175
GSS Graphics Development Toolkit....*CGI*	495	425
GSS GKS...*Kernel Sys,ANSI Level 2b*	495	425
Halo '88.......*from Media Cybernetics*	325	249
Microsoft Fortran..*Ver 4.0,inc. Codeview*	450	295
for XENIX	695	499
R/M Fortran...*ANSI 77 by Ryan McFarland*	595	425
for XENIX	750	599
Spindrift Library....*By Spindrift Labs*	149	129
PROLOG		
APT..*PROLOG Tutor*	65	59
Arity PROLOG Compiler & Interpreter	650	569
Arity PROLOG Interpreter	295	229
Arity Standard PROLOG	95	77
PROLOG-86 Plus.........*Solution Systems*	250	199
Turbo PROLOG...*Borland Intl.*	100	75
Turbo PROLOG Toolbox...*Borland*	100	75
TEXT EDITORS		
Brief......*from Solution Systems*	195	155
dBrief..*Macro lang for Brief & DBase*	95	75

	LIST	US
Brief & dBrief Combo	275	219
Condor Editor....*Condor Corp SUPER SALE*	130	65
Epsilon.....*Lugaru*	195	149
KEDIT.....*Mansfield..identical to XEDIT*	125	99
KEDIT Ver. 4.0	150	119
Pmate.....*Phoenix*	195	109
Vedit Plus.........*Compuview*	185	129
DEBUGGERS		
Advanced Trace 86.....*Morgan*	175	119
C-Sprite...*Source debugger for Lattice C*	175	139
Periscope I...*Board, Switch, Software*	345	289
Periscope II..*Breakout Switch & Software*	175	139
Periscope II.....*Software only*	145	105
Periscope III....*8 Mhz*	1095	875
Periscope III...*10 Mhz*	1195	975
Pfix 86 Plus....*Phoenix symbolic debugger*	395	199
LOGITECH		
MODULA-2 Compiler Package	99	79
MODULA-2 Development Pkg	249	199
MODULA-2 Toolkit	169	139
MODULA-2 ROM Package	299	239
MODULA-2 Window Package	49	39
PHOENIX		
Pasm 86......*Macro ASSEMBLER*	195	109
Pdisk......*Disk Management Utility*	145	95
PFantasy.....*six-pack take-away*	995	569
PFinish......*Profiler*	395	199
PFix 86 Plus.Symbolic Debugger	395	199
PForCe.......*Utility library*	395	199
PforCe...*Pforce for C*	395	199
PLink 86+...*sophisticated overlay linker*	495	269
PMaker.......*make utility*	125	74
Pmate........*Text Editor*	195	109
Pre-C........*Super-set of UNIX Lint*	295	154
Ptel.........*Binary Transfer Program*	49	39
POLYTRON		
PolyBoost....*Software accelerator*	80	64
PolyDesk III..*3rd Generation Desktop org*	99	73
PolyLibrarian...*Library Manager*	99	89
PolyLibrarian II	149	129
PolyMake......*Complete MAKE Utility*	149	129
PolyShell.....*UNIX-like Command Shell*	149	109
PolyXREF2...*Cross Reference Util all lang*	219	169
PolyXREF2...*Single Language support*	129	99
PVCS Corporate...*Source Code Control*	395	329
PVCS Personal...*Personalised ver of above*	149	129
PVCS Network...*Powerful Ver. of PVCS*	Call	Call
RYAN McFARLAND		
RM/COBOL......*ANSI 74 Standard*	950	675
for UNIX or XENIX	1250	999
RM/COBOL 85.....*ANSI 85 Standard*	1250	895
RM/FORTRAN.....*ANSI 77 Standard*	595	425
for UNIX or XENIX	750	599
RM/NET+ 5...*COBOL Networking*	300	249
RM/Screens.....*COBOL 85 Screen generator*	395	335
SOFTCRAFT		
Btrieve Softcraft's File Manager	245	175
Xtrieve...*Query language for Btrieve*	245	220
Report Option for Xtrieve	145	128
Btrieve/N File Management for Networks	595	445
Xtrieve/N.....*Multi-User Query*	595	459
Report Option/N...*Multi-user Rep Opt*	345	269
XQL.....*SQL for Btrieve*	795	595

HANSON & CONNORS'
PC EXCHANGE™*
1-800-DIAL-PCX
30 DAY MONEY BACK GUARANTEE

OVERNIGHT SOFTWARE DELIVERY
*Formerly PC EXPRESS

(That's 1-800-342-5729)
All prices subject to change.
Delivery subject to availability.

7 ZEL PLAZA
SPRING VALLEY, NY
10977

PHONE: (914) 426-2400
FAX: (914) 426-2600
TELEX: 667962 (SOFT COMM NYK)

CIRCLE NO. 233 ON READER SERVICE CARD

PROGRAMMING PRODUCTIVITY TOOLS

ESSENTIAL C UTILITY LIBRARY

400 Functions, 30¢ Each

You've probably seen the speed and power of Essential's C function library without knowing it. Software greats have been using it for some time to give today's top products pizazz and panache.

Now grown to 400 functions Essential produces pop-up menus, save and restore screens and windows to disk or memory in as little as 1/10th second, and the fastest video output available. Library has a complete set of 50 business graphics functions, 40 string handlers, 28 functions for printers, 18 for mice, 11 for time and date. DOS interfacing functions offer disk error trapping, directory and file management. Everything in source, including sample programs that demo library functions. We have versions with pre-built libraries for all well-known C compilers, and a source code librarian is supplied for rolling your own.

	List:	PC Express
C Utility Library	**$185**	**$119**
Essential Graphics	**$299**	**$225**
Essential Communications	**$185**	**$125**
with Breakout Debugger	**$250**	**$189**

C-TREE & R-TREE

B-Tree File Manager Now Has Report Generator

c-tree: The only major b-tree file manager with network support in the standard low-cost version. c-tree™ gives you record-locking routines for DOS 3.1/3.2, UNIX and XENIX, and it even comes in C source code, yet there are no royalties. Source sticks to K&R, so c-tree is portable. Tests in many environments prove it.

Permits any number of keys for a data file—alpha, numeric, even floating point. Handles files with varied record lengths, multiple keys in one index file. Both high level and decomposed functions. It's the works.

r-tree: Adds the ability to produce ad hoc reports from files maintained by c-tree (v. 4.1 and up). Link a file description to the r-tree™ library, and use any text editor to write report scripts with no further C coding. Reports can access data in several files, select on criteria, join findings into new logical records, sort them, calculate new fields and columns, tabulate by control breaks. Comes in source, same portability as c-tree, and fits any compiler.

	List:	Ours:	Combined:
c-tree:	**$395**	**$299**	**$499**
r-tree:	**$295**	**$235**	

WINDOWS for DATA

M'soft Windows Compatible

"Onlt one package can be easily recommended" said *Computer Language* (June '87) reviewing nine window and data entry products for C. Complete field level functions specify prompt string, field length, data type, screen location, picture, target variable, entry rules, help messages, even functions to call for validation once data keyed in.

Windows for C is a subset. No data entry but all windowing functions. Unlimited windows can be made either to pop up or permanently overwrite the screen, scroll and highlight lists vertically and horizontally. Specify Compiler. Windows for Data: List **$295**, Ours **$259**. Windows for C: List **$195**, Ours **$149**.

BLAISE C TOOLS PLUS/5.0

C TOOLS PLUS/5.0 from Blaise Computing Inc. helps you to quickly build professional applications using the full power of Microsoft C 5.0 and QuickC. Now you can concentrate on program creativity by having full control over DOS, menus, interrupt service routines, memory resident programs, fast direct video access; windows; printer and keyboard control, and more!

Blaise Computing's attention to detail, like the use of full function prototyping, cleanly organized header files, and a comprehensive, fully-indexed manual, makes C TOOLS PLUS/5.0 the choice for experienced developers as well as newcomers to C.

C TOOLS PLUS/5.0 prebuilt libraries are ready to use with either QuickC or the Microsoft C 5.0 command line environment. Complete documented source code is included so that you can study and adapt it to your specific needs.

	List:	PC Express:
C TOOLS PLUS/5.0	**$129**	**$ 99**
Turbo C TOOLS	**$129**	**$ 99**
C ASYNCH MANAGER	**$175**	**$135**
Turbo POWER TOOLS	**$ 99**	**$ 75**
Turbo ASYNCH PLUS	**$ 99**	**$ 75**

PANEL PLUS

Library Source Code Gives It Complete Portability

There are no end of tools for screen design and data entry, but none quite like Panel Plus. Design a screen under program control, use Panel's utility to "run" and test it field by field, then pass it to Panel's code generator which delivers C source code. Options style the code to your compiler's liking, and you can of course do what you like to the source afterward. The code calls Panel Plus's function library, but now *the library comes in source*, so everything produced is highly portable. Not like other screen managers delivered as object libraries and which leave *you* to write the detailed code.

Panel Plus will operate in graphics mode via interfaces to graphics products it supports and can utilize the EGA's 43-line screen. Low-level I/O functions adapt it to various keyboards, screens, operating systems.

Panel's newest incarnation has every imaginable feature. A single screen design can have 1000 fields stacked as visual overlays up to 127 levels deep or as pop-ups. Groups of fields can be moved between levels. Screens can be output as compilable code or stored on disk for loading at run-time. Each field can be boxed, colored, multi-row, word-wrapped, and scrolled horizontally and vertically if larger than its on-screen view aperture. It can be assigned its own help and error message, can be told to accept certain characters, or to match a picture, and to check data after entry—proper dates, number ranges, etc.—using Panel's or your own validation routines. You can add your routines to Panel's test utility because even it comes as source. Fields are accessed in any order and control reverts to your application program after each field for choice of action.

For past Panelists, the new version has smaller and faster field and screen functions, tighter granularity, and an enhanced, reworked library. Major tool for the serious developer. List: **$495**, PC Express: **$395**.

POLYTRON VERSION CONTROL

Source Code Control for Any Language

PVCS allows programmers, project managers, librarians and system administrators to control the proliferation of revisions and versions of source code in software systems. Independent programmers, the leading software publishers and LAN companies, and hundreds of Fortune 1000 companies rely on PVCS to store and retrieve multiple revisions of text. It maintains a complete history of revisions as an "audit trail", generates status reports, and uses intelligent "difference detection" to minimize disk space for each new version.

On Corporate and Network PVCS simultaneous changes to a module are merged into a single new version. If changes conflict, the user is notified. The "Logfiles" used to track changes are interchangeable between any PVCS product.

Corporate PVCS is for multiple programmers. It includes "branching" to maintain code when programs evolve on multiple paths. Personal PVCS offers most of the power and flexibility of corporate PVCS, but excludes multiple programmer features. Network PVCS is the Corporate version for LANs. File locking and security levels can be tailored to each project.

Ask for:	List	PC Brand:
Personal PVCS	**$149**	**$129**
Corporate PVCS	**$395**	**$329**
Network PVCS	**Call**	**Call**
PolyMake	**$149**	**$129**

dBC Identical dBASE III Plus Files Using C

dBC™ is a series of C libraries from Lattice which creates, accesses and updates files identical to those of dBASE itself. So dBASE can read and update the files too.

What for? It means both C and dBASE applications can operate on the same data bases interchangably. It means C programmers can interface with the big market of dBASE users out there, yet side-step the dBASE language. It means dBASE applications can now be linked to the universe of C libraries and tools to add windows, graphics, statistical analysis, all the things dBASE cannot do. It means the speed and power of C to impress clients accustomed to dBASE!

dBC's functions parallel all dBASE's file handling commands, many decomposed to permit direct data manipulation. Our versions of dBC mimic file formats for dBASE II and III and now dBASE III Plus makes your programs network ready!... as many stations as a network allows. Hands-off mode handles record and file locking and unlocking automatically. Close in functions give you direct lock/unlock control.

Supports all four memory models. dBASE II, III...List: **$250**, Ours: **$175**. dBASE III Plus...List: **$750**, Ours: **$595**. Call for Source Code Pricing.

Payment: We honor MasterCard, Visa, American Express (no surcharge), checks in advance, or funds wired to PC Express, c/o Chemical Bank, 126 East 86 St., New York, Account 034-016058. COD (U.S. only) for cash, money order, certified check (no fee). NY State, add sales tax. Purchase orders accepted from larger corporations and institutions at our discretion if you agree to net 30 days plus 2% a month late penalty thereafter.

Shipping & Handling: U.S.: UPS Surface: 1st product $6, each add'l $3. UPS 2nd Day Air: 1st product $12, each add'l $6. UPS Next Day Air and Federal Express shipment costs based on weight. International: Charges vary by destination and carrier. $10 per shipping container for export forms. Air parcel post at your risk beyond collected insurable amount.

C-WORTHY INTERFACE LIBRARY

The C-Worthy™ Interface Library wraps an entire user interface around your application. Its full power can be summoned by only a few high level calls. Sound exaggerated? A single function call can set up a complete text editor in a screen window. Recently acquired by Solution System, over 600 pages of Documentation, Turbo and Quick C version and a complete Interface Library have been added.

- High level calls pop menus and scrollable choice lists to the screen, restoring the background when dismissed.
- Windowing facilities open portholes of up to screen size for viewing virtual screens larger than the physical screen.
- Full context-sensitive help screen management takes over these chores and error messages. Automatic routines interrupt with pageable text windows explaining what to do next.

Novell found it "played a key role and accelerated development" in making its NetWare™ utilities easier for users. Ingenious demo: call for it.

Ask for:	List:	PC Express:
C-Worthy	**$195**	**$159**
with Forms Library	**$295**	**$269**

THE SPINDRIFT LIBRARY

Fully Functional Fortran Library.

Spindrift's smooth package offers something previously unavailable to the Fortran buffs...a basket of functions, packed to the brim. No more tedious coding. Just call on Spindrift with it's armoury of functions.

Any number of Windows may be defined, each with a striking border and brilliant color. Define "pop-up" screens, Save/Restore images, set Cursor shape. Snare an entire screen into a Character array.

The Keyboard; read it without echo, or sense any keypress during execution. Cursor controlled directly with Edit keys.

DOS interface includes: Call System, Call Exec, Findfrst/Findnext for (*) and (?) file searching. Search the Path, subdirectory and file manipulation, command line argument parsing, date/time functions, DOS environment access, and much more. Other utilities also included.

Specify compiler, Microsoft or Ryan McFarland. List: **$149.00**. Us: **$129.00**

PC EXCHANGE PROGRAMMER'S WORKBENCH

dBASE AT THE SPEED OF C

dBx Translates dBASE Applications to C

You dBASE™ programmers know what an expressive and readable language dBASE is. It's a very comfortable development environment. But the price is debased performance. Even compiled dBASE doesn't offer the speed that some users require these days. The kind of speed offered by software written in the C language. The answer is dBx™.

dBx translates dBASE to C. It offers you a major competitive advantage over the next dBASE programmer: Keep writing in dBASE. Take every application all the way to completion. Then use dBx to translate them top to bottom to C!

Other advantages: C is portable, even to other operating systems like UNIX/Xenix™. To the Macintosh or Amiga. dBx gives your applications a passport to places dBASE cannot go.

Has its own file manager for single user, but links to major C file managers—c-tree and dBC—for compatibility with dBASE files or multi-user support. We have everything you'll need, including good advice.

	List:	Ours:
dBx	**$550**	**$419**
with Library Source	**$950**	**$725**
Call for Full Source Pricing		

BRIEF/dBRIEF

The Power Environment for dBASE Programming

Many worthy utility products supply needs that dBASE®'s programming language doesn't—dUTIL™, dFLOW™ and a host of others. Trouble is, you have to use them separately, then combine their output into your dBASE program files.

No longer. dBRIEF™, written in BRIEF's macro language, grabs hold of BRIEF and turns it into a complete dBASE III and III Plus programming domain. Using BRIEF's underlying shell capabilities and its own interfaces, dBRIEF can run external utility libraries, plus dBASE itself, and link to the Clipper™, Foxbase+™ and Quicksilver compilers, all with dBRIEF still loaded and running the show. It can do what BRIEF already does plus:

- Convert a screen layout into dBASE code for interactive data entry.
- Display dBASE file structures in windows, a great convenience alongside your program files.
- Expand keystrokes into full dBASE statements.
- Indent automatically for clegic display.
- Create databases; index files; invoke Ashton-Tate's dFORMAT™ and dCONVERT™; draw lines and boxes.

"Simply marvelous programming environment for writing and editing dBASE programs.", *PC Magazine*, 7/86. Source code included!

Requires BRIEF 1.32 or later and 384k; 512k to run dBASE within dBRIEF; 640k and harddisk recommended.

BRIEF/dBRIEF...List **$275**, Ours: **call**

FOXBASE +2.10

The Fastest, Most Powerful, dBASE Compatible Database Management System, NOW!

New, **FOXBASE +2.10** is faster, and more powerful than dBASE IV...and it's available NOW! FOXBASE +2.0 is already a PC Magazine Editor's Choice...and has received an unbelievable 9.2 rating from Infoworld. FOXBASE +2.1 has new features such as built-in support for pull-down and pop-up menus, a new screen painter to create custom input screens, a new advanced automatic applications generator to write programs for you automatically, a new user interface designed to allow both first-time and experienced users to handle most database options effortlessly...and, much, much more... plus, Foxbase is still dBASE compatible and contains hundreds of dBASE enhancements. Run almost any dBASE program without modification. Foxbase fully supports commands like Browse, Edit, Append, Create, Modify, Report, and Modify Label, even in the middle of a compiled program! Foxbase supports EMS memory and local area networks.

	List:	Ours:
FoxBase +Version 2.10	**$395**	**$247**
Unlimited Runtime	**$500**	**$333**
FoxBase +Multi-User (LAN)	**$595**	**$397**
Unlimited Runtime	**$700**	**$467**
FoxBase +/386	**$595**	**$397**
Unlimited Runtime	**$500**	**$333**
FoxBase +Macintosh	**$395**	**$263**
Unlimited Runtime	**$300**	**$201**

MICROSOFT LANGUAGES

Powerful Implementations Of The Most Popular Programming Languages

Microsoft C 5.0: The flagship of the Microsoft line runs up to 30 percent faster than its predecessor. Its new optimization features deliver untouchable execution speeds, 100 new additional library routines...

Microsoft MacroASSEMBLER 5.0: If you ever wanted to take on the challenge of assembly, here's your opportunity. "MASM" 5.0 is a lot easier to use, has completely revised documentation, and a new "Mixed Language" programming guide that gives you step by step instructions for linking your assembly code with other Microsoft languages.

Microsoft QuickBASIC 4.0: is a revolutionary concept in BASIC programming. It allows you to run, edit, debug, and run again. Our friends at Microsoft have eliminated the dreaded compile step. Whenever you edit your code QB4 automatically incorporates your changes, so that it can run a program of 150,000 lines in less than a minute.

Each member of this language family includes the renowned debugger CODEVIEW.

	List:	Ours:
Microsoft C	**$450**	**$295**
Microsoft Macro-ASSEMBLER	**$150**	**$109**
Microsoft QuickBASIC	**$ 99**	**$ 66**
Microsoft FORTRAN	**$450**	**$295**

NOVELL: BTRIEVE, XQL, XTRIEVE

Sophisticated Tools Essential For Fast Database Handling

Btrieve is a library of subroutines that allows the programmer to build a database application using any language. It takes complete charge of all file creation, indexing, reading, writing, insertion, deletion, forward and backward searching. Its balanced tree indexing scheme finds any key in a million in less than 4 accesses...That's fast!

Btrieve is multi-lingual also. It includes more than 20 language interfaces (including C, BASIC, PASCAL, FORTRAN). However if it turns out that you are using something a little unusual, worry not. The manual includes a chapter on how to write a language interface to Btrieve.

Btrieve's vital statistics are equally impressive. Files may have up to 24 indexes; fixed record length to 4090 characters; variable length to 64K; indexes to 255 characters; files of 4 billion bytes. Network support includes Novell, 3-COM, IBM PC NET, Software Link's Multilink and many others.

XQL is a relational database management system designed especially for programmers. Imagine being able to access your database with the ease of SQL (Structured Query Languege) statements and still having the power to process that data right down to the byte level.

Think about your applications. A large part of your software development effort is probably devoted to managing data stored in files on disk. Hours spent writing lines of code to search and store data records could have been used to program more important parts of your application. Why not let XQL do it for you. XQL will increase your programming productivity and let you focus on building better applications.

The XQL system works in tandem with Btrieve and has an equally powerful chassis...No limit on the number of records per file. Max. file size is 4 gigabytes, Max. record size equals 4K, Max. indexes per file is 24. The one version works for single or multiuser systems, DOS Ver 3.0 or greater. All languages are supported.

Xtrieve is the final ingredient in the Novell programming recipe. It is a menu driven, data retrieval system, that allows you to quickly find information and display reports. System developers can easily customize Xtrieve to display command menus, help files, and error messages in the English spoken by the customer. Xtrieve screens then gives menu choices that users can quickly recognize, making Xtrieve an easy product to use and understand.

Report Option for printing customized reports, form letters, mailing labels & statements.

	List:	Ours:
Btrieve	**$245**	**$175**
Btrieve/N	**$595**	**$445**
XQL	**$795**	**$595**
Xtrieve	**$245**	**$220**
Xtrieve/N	**$595**	**$459**
Report Option	**$145**	**$128**
Report Option/N	**$345**	**$269**

GSS GRAPHICS SYSTEM

Leave the Device Driving to GSS

For serious applications stick to the tools that stick to the standards. Not the least of reasons why GSS™ has emerged as the pre-eminent graphics toolmaker is that it has always conformed to ANSI standards.

At the heart of the system is the CGI standard Development Toolkit. It has all language interfaces and device drivers for keyboards, mice, joysticks, tablets, printers, plotters, cameras. The drivers completely insulate your application from concern for device idiosyncracy.

GSS Kernel™ conforms to ANSI's GKS 2b and has all its drivers and language bindings. Macro level tools to draw, color, segment, transform, store and recreate an object. The Metafile Interpreter reads ANSI CGM files with full CGI capability for recreation on various devices.

Quality software? IBM thinks so. They sell GSS under their own label. Royalties. Needs 256k.

	List:	Ours:
CGI Dvlpment Toolkit	**$495**	**$375**
Kernel System	**$495**	**$375**
Kernel for IBM RT	**$795**	**$645**
Metafile Interpreter	**$295**	**$235**

CLIPPER $399

R&R RELATIONAL REPORT WRITER $129

MORE PROGRAMMING TOOLS FOR...

dBASE, C, FORTRAN, BASIC, COBOL, PASCAL

dbc ISAM, FoxBASE, Greenleaf, Halo '88, Logitech MODULA, MicroFocus COBOL, Norton Utilities, Panel Plus, Periscope, Phoenix, Polytron, RM/COBOL, Sprindrift, Tom Rettig's Library, Windows for Data...and many more.

Call or write for prices and information.

HANSON & CONNORS'

PC EXCHANGE™* **1-800-DIAL-PCX** **30 DAY MONEY BACK GUARANTEE**

OVERNIGHT SOFTWARE DELIVERY
*Formerly PC EXPRESS

VISA MasterCard AMERICAN EXPRESS

(That's 1-800-342-5729)
All prices subject to change.
Delivery subject to availability.

7 ZEL PLAZA
SPRING VALLEY, NY
10977

PHONE: (914) 426-2400
FAX: (914) 426-2600
TELEX: 667962 (SOFT COMM NYK)

Little Big D

A speedy little data manager with a little price and a lot to offer, D The Data Language challenges an established market by minimizing code to enhance productivity.

VICTOR E. WRIGHT

With so many PC data managers already securing large portions of the market (led by Ashton-Tate's dBASE III PLUS), it takes courage to introduce a totally new product. But that is what Caltex Software Inc. of Dallas, Texas, has done with D The Data Language 2.7.

D is a little product with a little price ($395) and a lot to offer, including an English-like fourth-generation programming language (4GL). 4GLs require fewer, simpler statements to do more work than other high-level languages. As a 4GL, D has elements that are both procedural (requiring the developer to specify how results are to be produced) and nonprocedural (allowing the developer to specify results without regard for methods).

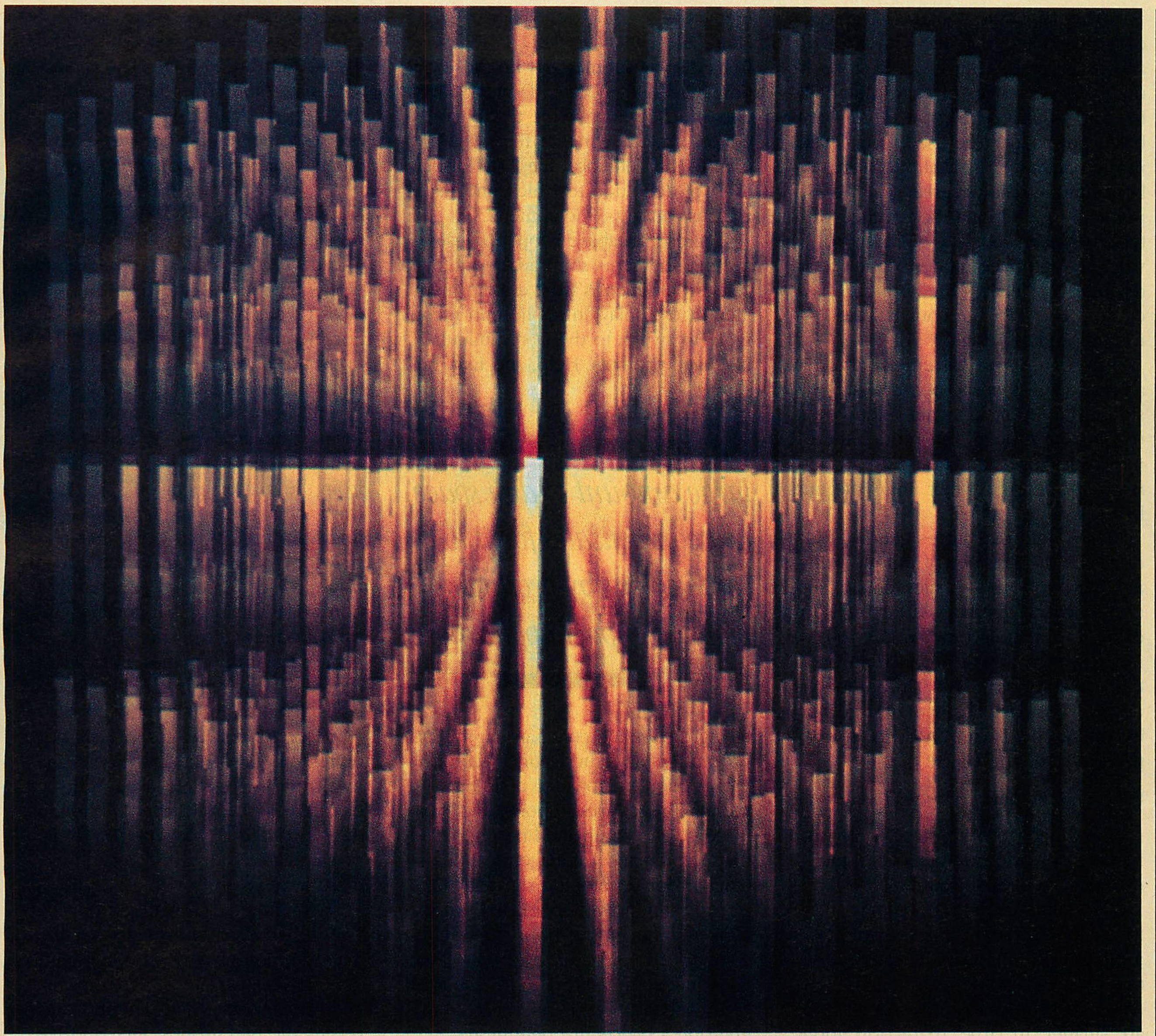
Caltex calls D a better-than-4GL (or 4GL+) development language because its procedures for handling data—selecting (isolating), arranging, and reporting—mimic real-world procedures and go even further than 4GLs to reduce development time and increase productivity. If this is the test of a 4GL+, then D deserves the title.

D's development speed comes from its terse 84-command language (also called D), which makes it easy for developers to perform operations using just a little code and a few statements. The terseness also minimizes the learning curve, the likelihood of errors, and the debugging time. The language is the only one needed for both development and retrieval.

Further, D eases the task of creating the application once the database is designed. Even though it has no application generator (which automatically generates applications code from developer specifications), D requires about the same effort as systems that have them. This is because code generators can produce vast amounts of code that the developer must compile and debug, while D *remembers* screens so that developers need not write any code to create or manipulate them.

D also offers great performance. Although it does not have the fastest times based on *PC Tech Journal*'s benchmarks, D ranks with the best.

D's ease of use—users don't have to be programmers—and its moderate price combine to make it an excellent choice for many small companies. Developers of data-management systems will find D an excellent platform on which to develop applications targeted toward small business as long as D's considerable disk-space requirements, which depend on the size of the database, are acceptable. Because this product is not yet available in a multiuser version, it is of limited value for many



COMPUTER GRAPHICS • DOV JACOBSON

big businesses. Caltex is working on DOS network and Xenix 286 versions of D; the schedule for developing an OS/2 version is not firm.

DEEP IN THE HEART OF D

D is a data manager based on the relational model. It meets many of the criteria developed by E. F. Codd to define relational data managers (see "Lingua Franca for Databases," Richard Finkelstein, December 1987, p. 52).

D organizes data into tables called data groups; retrieves data by specifying a combination of tables, primary key values, and fields (columns); supports data definition, view definition, and data manipulation; adds or deletes data based on a table or view (an image of a record set selected from one or more tables); and makes integrity constraints part of the database. D does not support null values; instead, it assumes that no input means zero in number fields and blanks in character fields. D does not include the relational operators by their textbook names, but its commands perform the relational functions (selection, projection, product, union, intersection, difference, join, and division).

A D database has one or more data groups with fields of allowable data types—alphanumeric, numeric, long and short integers, and single- and double-precision floating-point numbers. D stores integer and floating-point numbers in binary form. A record's combination of field lengths cannot exceed 4,088 characters.

Relations among data groups do not need to be defined explicitly in the database definition to establish procedural links in applications. Rather, D defines one-to-one, one-to-many, and many-to-many relationships by providing common field definitions in two or more data groups. Records in two or more data groups are related if their common fields contain identical values.

The developer defines the database in a text file with no file extension. D compiles the file to produce six database files: a control file (.DBD); a data file (.DDD); a tree-structure file (.TTT); a list-structure file (.LLL); a general-interface file (.GI); and a user-application catalog (.PRS), which stores every menu, procedure, report, and screen definition for an application.

The .DBD file includes a compiled definition of the database, which describes actual records in the data file and index structure. The .DBD is not manipulated in any way during applications development or database updating, thus maintaining its integrity. This avoids damage to the database definition through database operations.

The .DDD file contains data records stored sequentially, with a time and date stamp. The .TTT file stores unique occurrences of data in an .LLL file known as a *cellular multilist*—a balanced, binary tree used to navigate in the database. The .LLL file stores both unique data occurrences and pointers to additional occurrences (called *synonyms*). The .GI file provides the interface name to open a database and names the application catalog. The .PRS file contains the user-application catalog.

D requires an extremely large amount of disk space because it assumes that every key data value is unique. Developers can reduce the index-structure size by adding a clause to a field definition that limits the number of unique values.

In a field that stores gender, for example, the number of unique values is limited to three—male, female, and not known. If gender is a key field in a data group that accommodates 100,000 records, then D will produce a tree that accommodates 100,000 unique values. A large database with a large number of key fields can produce an expansive tree. The clause

Unique <field name> *xx* <field name> *yy*

reduces the size of the tree to accommodate only the number of unique data occurrences specified.

To some extent, D compensates for its storage requirements by producing highly compact applications. Although the database itself may be larger than other systems, applications are smaller. In large, real-world applications, D's storage requirements compare with those of other systems.

VIEW FROM OUTSIDE

Although D's interface does not at first appear state-of-the-art, it is actually well designed. Developers can select most development functions quickly using point-and-shoot menus or by directly typing D language commands. At the master menu, pressing an arrow key displays and positions a light bar; then pressing the Enter key selects the highlighted option and D prompts for any required parameters. At the command prompt, the developer can key in options with parameters or without, in which case D prompts for them.

D's master menu selections are Open (to open a database that has been compiled earlier), Dcat (to display a list of compiled databases), DBD (to display the database-definition menu), PI (to display personal-interface menus for specific database views), Help, and Quit. The DBD and PI selections display submenus with these options: Old, New, Comp, Dcat, and Exit. Old and New invoke the text editor to edit or create definition files; Comp compiles a definition file; Dcat displays a list of definition text files; and Exit returns to the previous menu.

After a database is opened using the master menu, the developer can select four options from a main menu: Menu (to display an application menu), DCL (to switch to the D command-language prompt), Help, and Exit. Entering the Define command at the DCL prompt displays the Define menu, which offers options for defining menus, procedures, reports, and screens. The command-line interface to the D command language, or DCL, is efficient for developing short sections of code interactively because D maintains a command history review. A developer can work out a section of code, enter the editor, and recall the command history for reference while entering a procedure definition.

D requires an extremely large amount of disk space, but compensates for it to some extent by producing highly compact applications.

Developers can customize end-user interfaces by incorporating all elements of D's interface into an application. They can call both full-screen and pop-up, point-and-shoot menus from procedures or other menus. Developers do not have to follow a structure but can place items anywhere on the screen. This gives the developer substantial control over the appearance of an application's user interface.

When the developer defines a database, D automatically produces the general interface and application catalog. The developer can place additional menu, screen, report, and procedure definitions in the catalog. By creating an unlimited number of *personal interfaces* for specific database views, the developer can limit user access to only the data required by the application.

D's interface makes excellent use of function keys. They display help screens, zoom into lists associated with screen-transfer fields, access dialog history (a log of every command typed on the screen during a D session), and terminate the current task. F1 enters the help mode, and F9 returns to the previous menu. Function-key assignments cannot be modified.

A JOY FOR DEVELOPMENT

Application development in D is easy. The developer defines a database and writes procedures and personal interfaces (if desired), all using the DCL. For menus, screens, and report definitions, the developer can use the defaults or customize them using any text editor, including D's own editor.

Defining the database. The developer defines a database by selecting DBD from the master menu or selecting DCL from the main menu and typing "Define Database" at the D command prompt. The resulting database-definition file (see figure 1) is the application's data dictionary. It contains all information about the database—its name, data-group names, number of records in each group, field names, types, lengths, and key field indicators.

Developers can group data-field definitions into data structures using level numbers. A field name followed by a series of lower-level field definitions is called a *grouping field* and provides the equivalent of multisegment indexes and overlapping fields found in other data managers. In *PC Tech Journal*'s data-manager benchmark application, the grouping field AUNAME appears in both the Author and Article data-group definitions and relates the two groups. First and last names are available through the individual fields, FNAME and LNAME. (For a complete description of the sample application, see "Evaluating Data Managers as Development Tools," Julie Anderson, August 1985, p. 46.)

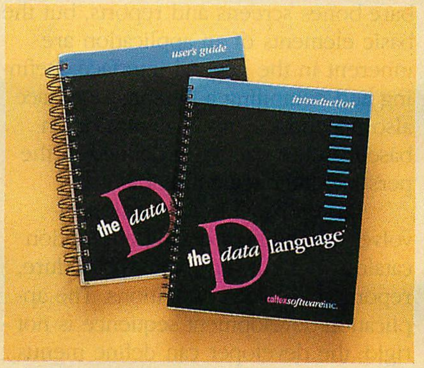
Each field definition includes a level number with 0 being the highest level. Lower-level field names are preceded by a colon and a level number. A field name that is followed by one lower-level field definition can create alias field names. Using the grouping field name in several data groups and a local alias in only one group permits a single field to be both *global* (known in two or more groups) and *local* (known only to the current group).

After creating and saving the database-definition file, the developer compiles it to produce the general interface, personal interfaces, and application catalogs. The source file remains intact for editing and recompiling.

D THE DATA LANGUAGE OVERVIEW

D THE DATA LANGUAGE
Caltex Software Inc.
3131 Turtle Creek Blvd., Suite 1101
Dallas, TX 75219
214/522-9840
CIRCLE 334 ON READER SERVICE CARD

Product description: D The Data Language is a relational data manager and fourth-generation language designed to provide the developer or end user with a high-productivity environment for developing both simple and large, complex database-management applications.
IBM PC environment: D runs on the IBM PC/XT/AT, PS/2, and compatible microcomputers with 512KB of RAM, a hard-disk drive, one diskette drive, DOS 2.0 or later, and a color or monochrome monitor. Monochrome display of graphs and pie and bar charts is available on machines that support the CGA all-points-addressable mode. D automatically detects and uses a math coprocessor.
Network support: D does not yet run in multiuser environments.
Copy protection: None.
User interface: The user interface provides point-and-shoot menus, a command line, and function-key support. Applications can include full-screen menus, pop-up menus, interactive screens, and command prompts.
Documentation: Caltex provides an *Introduction* manual and a *User's Guide*. The *Errata* will eventually be merged into the other manuals.
Help facilities: On-line, context-sensitive help is available at a keystroke in all development modes and during application execution. The developer can incorporate custom help screens.
File capacities: Records are limited to 4,088 characters, regardless of the number of fields. A database can contain an unlimited number of data



groups. A database or data group can contain an unlimited number of records, subject to operating-system file-size limitations. Any field in a data group can be a key field; only the first 60 characters are used for the purpose of indexing.
Field types and capacities: Alphanumeric, 4,088 characters per field; numeric, 4,088 digits, not including the decimal point; short integers as large as 32,767; long integers greater than 32,767; single-precision floating-point numbers, up to six significant digits; and double-precision floating-point numbers, more than six significant digits.
Data entry: D supports full-screen data entry and validates data type and field lengths. Developers can provide additional validation using menu-defined field services associated with data-entry fields. Special procedures attached to fields can perform other data-validation procedures.
Application development facilities: D provides a structured programming language that includes If . . . Else . . . Endif, Goto . . . Label, Exit, and End statements; nested procedures and menus; built-in arithmetic, financial, string, trigonometric, date, and type-conversion functions; field statistics; and global variables.
Security: Multiple data views can be defined using personal interfaces to limit access to the database.
Access to system facilities: Commands exist to change and print directories, erase files, and control the printer on-line status. A DOS shell is not provided, but external programs incorporating a host-language interface can be called by procedures or menus.
Querying and sorting: Queries are accomplished in three steps: isolate (to place copies of records in collections stored in RAM or temporary disk files), arrange (to place isolated records into the desired order), and report (to produce printed reports and screen displays). Queries can isolate records from multiple data groups using Find, Fill, Look, and Match.
Reporting: Reports are displayed using a Print command or a report definition called with the Report command. The default report format is a simple columnar report with field names as page headers, but reports can be customized. Field statistics, including average, maximum, minimum, count, sum, variance, and standard deviation, are maintained automatically and can be printed at detail, breakpoint, and end-of-report level.
Utilities: D is self-contained. Database maintenance and recovery are done with commands or procedures written in D. Caltex also provides a text editor and help-file generator.
Data compatibility: D can read and write .DIF, fixed-length ASCII, delimited ASCII, blocked data files, and .DBF (dBASE II and III PLUS) data files.
Distribution: D is available directly from the vendor and through dealers.
Price: $395.
Support: Caltex offers telephone support and annual maintenance agreements for D.
—*Victor E. Wright*

At the end of compilation, D offers four options for initializing the database: (1) create a new database by acquiring and initializing new files large enough to hold all records specified in the database-definition file; (2) modify an existing database by updating the .DBD control file to the new definition and resetting the structure files (.TTT and .LLL) while preserving the data files (.DDD); (3) delete all files and replace them with new ones; or (4) return to the .DBD menu.

A D database cannot be restructured without recompiling it. In contrast, fully relational SQL-based packages allow developers to insert new tables in the database at anytime. In a future release of D, Caltex plans to add an SQL layer to allow inserting new tables or fields without having to recompile the database. D currently provides a Key command to declare a field as a key field without recompiling the database definition, so the developer can add keys anytime.

A language all its own. D's command language is for both application development and ad hoc queries from users and developers (see table 1). It allows the structuring of applications into procedures and menus.

Procedures can call other procedures and menus; menus can call procedures and other menus. Within procedures, the constructs If . . . Else . . . Endif, Goto, End (terminates the procedure), and Exit (returns to the main menu) provide program-flow control.

FIGURE 1: *File Format*

```
Database name is PCTJBNCH

Dg author 1000 records

  Auname              key
  :1 Lname    X(18)   key
  :1 Fname    X(12)   key
  Address
  :1 Street   X(20)
  :1 City     X(16)
  :1 State    X(2)    key
  :1 ZIP      X(5)
  Work_phone  X(10)
  Home_phone  X(10)
  SSN         X(9)
  Biography   C(200)
```

Databases are defined in any text editor including D's. The developer names the database and data groups, states the maximum number of records expected, and defines all fields.

The developer can create variables with the Set, Fetch, Calc, Choose, and Prompt commands. Set assigns a value to a variable directly; Fetch transfers data from an isolated record to a variable; Calc assigns the result of a calculation to a variable; Choose assigns the result of a menu selection to a variable; and Prompt assigns a general-user response to a variable. Developers can use variables, which are all global, in commands, procedures, reports, and menus and can display them using the Vlist command. The default configuration imposes a limit of 50 string and 50 numeric variables. The developer can raise or lower these limits by using the configuration file, CONFIG.D.

D maintains three system variables called *answer-cell variables*—Uncalc, Unset, and Quit—that automatically fill with commands to indicate a completion status and/or error codes. Uncalc and Unset erase variables created with Calc and Set. D maintains these variables until the developer enters the Quit command at the master menu. Variables created and bound by one application can pass values to other databases or applications in the same database, within a single D session.

Although a slight speed penalty may result because D interprets procedures, it is offset because D compiles report definitions on the fly immediately before displaying reports. This executes procedures immediately after they are defined, which is certainly an advantage in application development.

Debugging tools include a command history, a Diagnostics command that displays current error messages, answer-cell variables, and the Vlist command that displays active variables.

Customizing with flexibility. D automatically generates default data-entry screen and report definitions. They produce bare-bones screens and reports, but the basic elements of an application are inherent in the general interface. Defining and compiling a personal interface also produces a minimal application based on the data view defined in the personal-interface definition.

Customizing an application involves extending a default application catalog with custom menu, procedure, report, and screen definitions. The application-development sequence is not rigid; the developer can define menus, procedures, reports, and screens in any order and modify them anytime.

D allows customization of menus. To create a menu—which can have the form of selection letters or numbers followed by explanations or text—the developer enters text and line-drawing characters with any editor and indicates the names of procedures and submenus using the F5 key. The names of procedures and submenus to execute from the menu must appear on it. Once the developer saves the menu definition, it is ready for use. At the menu, pressing the up or down arrow keys displays a light bar that moves from one marked word to another.

D's menu-definition process is easily as fast as wading through a series of menu-generator screens and is faster than typing a series of *@xx,yy* Say . . . statements. The Define Menu command does not generate code—it simply accepts the definition of a menu. D takes care of displaying the menu, displaying and moving the light bar, accepting user input, and executing selection without the need for programming.

Full-screen entry and update. D supports full-screen data entry and update. The Append command adds records, and the Update command modifies existing records in a data group or groups. Options for both commands include automatic advance when a field is full, automatic return after appending a specified number of records, insert instead of typeover, overriding data-validation error messages, suppressing automatic key updates, and specifying the type of relationship that applies to the update session (one-to-one or one-to-many). In addition, the developer can assign default values to fields by including assignment statements of the form <field name> = <value>, in the Append statement.

D provides default data-entry screens for appending, updating, and reading records. Screens in D consist

TABLE 1: *D Commands*

ADD, CHANGE, DELETE DATA	
Append	Raze
Change	Undelete
Delete	Update
Load	Write
ARRANGING	
Order	Sort
Relate	
CATALOG MANAGEMENT	
Cat	Purge
Copy	Undef
Dcat	Xchange
Pack	Xref
CREATE APPLICATIONS	
Define	Redef
FILE IMPORT/EXPORT	
Load	Unload
Report	
INFORMATION	
Codes	List
Config	Tally
Help	Vlist
Info	
ISOLATION	
Clear	Match
Fill	Pick
Find	Reduce
Look	
MISCELLANEOUS	
Beep	Pagespec
CD	Poff
Checkpoint	Pon
Color	RS
Crtspec	Space
Diagnostics	Timing
Dir	Uncalc
Erase	Unset
Open	
REPORTING	
Print	Report
Rawprint	Show
Read	
STRUCTURE MANIPULATION	
Construct	Key
Destruct	Structure
TRANSFER CONTROL	
Call	Goto
DCL	Menu
End	Mode
Exit	Quit
USED IN PROCEDURES	
Calc	Inc
Choose	Pause
Cls	Prompt
Fetch	Set
If	Type
Iff9	Verify

Only 84 commands are needed in D for complete data definition, data manipulation, and program control.

of *labels* to prompt for data and *transfer fields* to accept data (see photo 1). When the user enters data, the cursor parks in each transfer field to accept them. The right and left arrow keys move the cursor within a field, and the up and down arrow keys move the cursor from one field to another. PgUp and PgDn move through screen pages. The developer can customize screens with line-drawing characters, screen attributes, and color using a text editor.

At the default screen, D validates only the field type and length. Custom screens can specify more extensive data-validation tests using the field-services utility. Each transfer field has an associated field-services menu that displays when the developer defines or edits that field (see photo 2). Data-validation options on the menu include Auto Repeat, Auto Increment, Current Date, Required Field, Must Fill, Template (picture), Length, Decimal Point, and Force Uppercase.

Some of the conditions that developers can specify are Unique, Exist, In Range, In List, Alpha, Numeric, and Alphanumeric. When Must-be-in-list and Must-be-in-range are toggled to Yes, fields appear in which the developer can specify the list and range.

Developers also can verify data against a set of records, which must be created before invoking Append. This allows verification against a dynamic group of values, as opposed to static groups specified by lists and ranges.

When the developer tags fields with the Exist or List conditions, the F3, F4, and F5 keys display allowable values. F3 steps through the list from the beginning, F4 steps through from the end, and F5 displays a pop-up menu from which the developer can select values using the light bar.

Also using the field-services menu, developers can attach special procedures composed of a subset of D commands to fields (see table 2). These procedures can perform data validation that other field-service options cannot, such as calculating derived values for other fields, passing messages to and from the user, displaying pop-up menus and obtaining responses, controlling cursor movement, and creating or modifying variables.

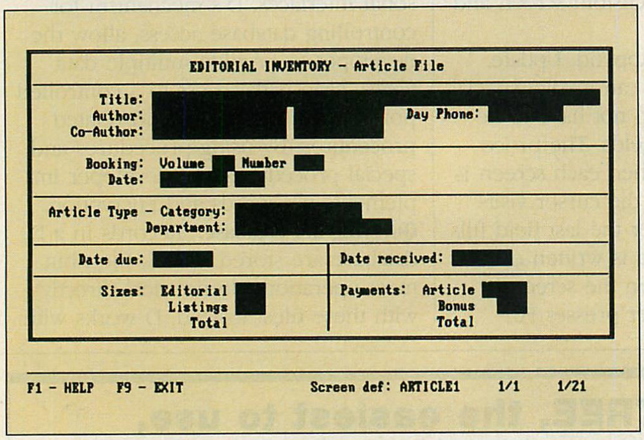
In order to append records to multiple data groups simultaneously using the Append command, the developer must define a custom screen because D creates a separate default screen for each data group. To update records in any one data group with the Update command, a single default or custom screen is sufficient. Updating multiple data groups from one screen, however, requires a custom screen and special procedures.

Options to the Append, Update, and Read commands can invoke special procedures, which do not have to be attached to specific fields. The procedures execute: (1) when each screen is displayed but before the cursor visits the first field; (2) after the last field fills but before the record is written or updated; and (3) when the screen is displayed and the user presses F6.

Passwords and data-encryption commands are not built into D. Personal interfaces, D's mechanism for controlling database access, allow the developer to provide multiple data views, each with access to a controlled portion of the database and related procedures. By using procedures and special procedures, the developer implements passwords and encryption.

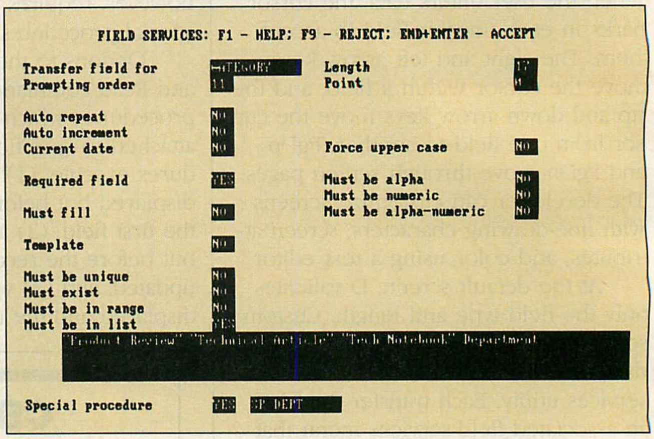
Querying the database. Records in a D database are stored in disk files, but most operations do not deal directly with these files. Instead, D works with

PHOTO 1: ***Sample Data-entry Screen***



Custom data-entry screens are developed by typing text prompts, marking transfer fields to receive input, and painting lines. D will take care of details, such as displaying a light bar and positioning the cursor on transfer fields.

PHOTO 2: ***Field Services***



Pressing F5 while defining a data-entry screen in D's text editor displays the field-services menu from which developers select validation methods such as Length, Must fill, and Range, or attach special procedures to each transfer field.

collections, RAM-resident images of the database supplemented with a temporary disk file if insufficient memory exists to hold the whole collection.

The three phases for constructing queries and reports are isolate, arrange, and report. The isolate phase creates a collection containing an image of the entire database, a single record, or a set of records selected from one or more data groups or the entire database using values of key fields. The collection is in the order in which records enter the database. The arrange phase sorts records, and the report phase produces both screen displays and printed reports.

The default configuration of a database includes five collections named A, B, C, D, and E. Using a personal-interface definition, the developer can rename these default collections and add extra collections having arbitrary names up to 12 characters long.

Central to D is the set of commands, including Find, Match, and Look, that isolate data into collections. Find has the syntax

```
Find[/option] <collection> [<dgname>]
  Where <query-expression>
```

Find searches for all records that meet the conditions of the query expression. This command places images of the records in record-number order into the collection. The query expression can contain one or more simple expressions, each consisting of a field name, a value, and a relational operator such as >, <, and =.

Find acts like the relational algebraic operator Select, placing entire records into the collection. It is not like SQL's Select, which is the equivalent of the algebraic Select followed by Project. D's database engine is so slick, however, that entering only a collection name and a query expression can create a collection, without having to use the Find command.

The Match command expands a collection isolated by Find. Using equal values in common fields, Match effectively isolates records related to records already in the collection. Although similar to the relational operator Join, Match does not actually combine related records to form a new, wider table; it leaves records intact for the developer to arrange.

Look, a full-screen equivalent to Find, displays the default screen or a named screen with blank transfer fields. The developer forms the query expression by entering values in transfer fields. After pressing the Enter key from the last field, D displays the first record that matches the query expression. If the developer limits expression to naming fields in the command, the cursor visits only named fields and the system selects records when the last named key field is filled. When the Look command is ended with the F9 key, the collection remains for other uses, such as report writing. Although Look is not query-by-example, it displays the database and allows users to select records.

In many cases, Find isolates records that have duplicate values in some fields—a mailing-list isolation might include several members of some families, for example. The Reduce command removes records with duplicate values in specified fields.

Isolation commands operate only on key fields. A complex query expression, however, can isolate records from multiple data groups, whether or not the groups have common fields. Other isolation commands include Fill (which isolates the entire database or an entire data group into a collection), Pick (which isolates records by record number), and Clear (which empties a collection). After isolating the data, the developer can arrange them in a collection using the Sort, Relate, and Order commands.

The developer sorts records from only one data group in the collection by specifying a data group with the Sort command; from multiple fields by specifying field names; or from the entire collection by default. D sorts records that do not have specified sort fields if they have high or low sort-field values, indicating that they go at the beginning or end of the sort.

Developers can report contents of arranged collections by displaying or printing them. A report may be *comprehensive*, listing the entire contents of the collections, or *limited*, listing a subset of fields available. A single arranged collection can produce material ranging from simple columnar reports to multilevel indented invoices.

The Define Report command displays a menu with selections to create a new report, edit an existing report, or display the catalog of existing report definitions. Selecting New or Old invokes D's text editor. The developer uses the editor to create or edit the report definition by entering report-definition statements, such as Return, Every, and Calc.

The report definition contains four sections: Detail, Breakpoint, Atend (at end), and Titles and Declaratives. The Detail section begins with a Return statement and contains a description of the detail lines of the report. Breakpoint begins with an Every statement and contains statements that produce breakpoint summaries and statistics. Both sections can contain Calc statements and If . . . Endif statements. The Every statement can name several fields as breakpoints. When a breakpoint-field value changes, the statements following the Every statement execute. These statements can print subtotals, summaries, statistics, and so on.

The Atend section begins with the Atend statement and specifies tasks to be completed at the end of the report, including calculations, printing totals and statistics, and updating a data group in the database. The Titles and Declaratives section contains statements to control format of the printed report. This includes producing a report title, page headers and footers, section headers, and titles embedded in the report.

D's default report is a simple columnar report with field names for column headings. To produce it, the developer enters the Print statement at the DCL prompt. D then saves its definition in _PRINT, which can be copied to form a named report definition. Although the default report is simplistic, it is an excellent prototyping tool.

The data manager produces reports, like queries, by isolating, arranging, and reporting data. The first two phases are essential to producing reports efficiently; complex reports may not be correct if the isolated data are not arranged properly. Report definitions generally are hidden inside a procedure that performs isolation and arrangement.

Graphics reports include pie charts, bar charts, and histograms displayed on the screen. These reports can be printed only by using a graphics screen dump.

For reporting, D has access to a system variable, %Level, which tracks the level of the report writer in a hierarchical arrangement of records from several data groups. Used in an If statement, the %Level variable allows the report writer to perform tasks when a given level record is read. Similar in concept to the Every statement of a breakpoint, the %Level variable allows more complex tests.

The Type and Pon (printer on) commands each send printer-control codes to the printer. The developer also can send these codes by embedding them in any line of text going to the printer. The format is the commonly used \xxx, in which *xxx* is a decimal number.

TABLE 2: *Special Procedures*

DCL COMMANDS FOR SPECIAL PROCEDURES	
Beep	Fill
Calc	Find
Call	Goto
Change	If
Checkpoint	Inc
Choose/p	Menu/p
Clear	Pick
Color	Rem
End	Set
Exit	Timing
Fetch	Write
COMMANDS FOR SPECIAL PROCEDURES ONLY	
Cursor	Reject
Move	Select
Msg	

Developers can use a subset of D commands to write special procedures to perform data validation, calculate derived values for other fields, pass messages, and display pop-ups. These special procedures will attach to any transfer field on a screen.

Crash landings. A crash in D results in a possibly corrupt, database structure. Generally, no more than one record is lost because D writes to the data file for every update operation. Still, D assumes the tree and list files are corrupt because buffers that retain index structure normally update only when the database closes properly (non-crash conditions). D opens the database with the /D (down) option and warns that the structures could be unsafe and new data could be lost.

If the session during which the crash occurs does not involve writing to the database, all may be well. It is safe to assume the index structure is corrupt; take corrective action by destroying the old structure with Destruct and rebuilding it with Structure. D protects database integrity by requiring this sequence of operations from within the general interface. The Checkpoint command, which flushes all buffers to disk, can minimize damage to a structure. Following each Append or Update command with a Checkpoint command is cheap insurance.

Documenting D. Documentation consists of an *Introduction*, a *User's Guide*, and an *Errata*, a booklet describing additions since the manuals were printed. The *Introduction* is well-organized and thorough, covering all D operations from queries to creating applications. It includes a tutorial for installing D; using D's text editor; creating a database, querying and reporting it; creating custom screens; and installing D's example database.

The *User's Guide* has sections on database definition, personal interfaces, applications, report definitions, database maintenance and recovery, and an alphabetical command reference. The *User's Guide* and the *Introduction* are both sparsely indexed. Caltex offers an annual maintenance agreement and telephone support.

DELIVERING TO END USERS

Applications written in D can be distributed with a full-development system or with a runtime system. Those distributed with the full system can allow complete access to all D databases; with the runtime system, users can perform only functions allowed by the developer to a single database. The runtime product omits all commands that modify the application catalog, such as Define, Redef, Copy, and Xchange. The developer can allow the end user to have ad hoc query capabilities with the Mode Stop command.

Installation. D's installation program copies files to the hard disk and performs housekeeping for proper operation of overlay files. The product cannot be installed by simply copying the distribution diskettes to the hard disk.

Assorted utilities. D's text editor is full screen. Although any text editor can create database-definition files that can be imported into D using the Copy command, only D's text editor is invoked from within D. Although D's editor is not a full-fledged word processor, it has all facilities to create and edit source files, such as text entry, line drawing (using the PC's extended characters), and assigning screen attributes (such as highlight, blinking, and reverse video). The text editor can move the cursor by character, word, or screen; split and join lines; and delete, move, and copy blocks of text. Its commands are keystrokes or key combinations, but an Edit Command mode replaces or copies text.

D's extensive on-line, context-sensitive help facility available during both development and execution of applications, is one of the best among currently available data managers. Two alternatives exist for obtaining help. First, at the DCL prompt, typing "Help"

provides a summary of the help system and entering "Help <command>" displays a help screen for the named command. The second alternative is to press F1 while entering a command to split the screen and display help messages on the lower portion. In D's editor, pressing F1 displays help screens related to the text editor itself.

D includes a help-file generator so that any third-party text editor can create context-sensitive help files that operate like D's help file. The help-file generator, HF, compiles each help file, consisting of a series of help screens and an index. A file reference goes into the corresponding personal-interface file, which is then recompiled. At the DCL prompt, typing "Mode Help <screen name>" links help screens to sections of an application. The current help screen is set by placing Mode Help commands in procedures.

For generating screen or application code, D has no utilities. The developer must write the three basic elements of an application—the database definition, application code, and report definitions—line by line. However, this lack of generators is not a real downfall because of the brevity of the language, the quality of the on-line help facility, and the fact that D defines screens without code. Developers also must key in procedure and report definitions, but that, too, moves fast in D.

To set environment specifications, global parameters are defined in CONFIG.D, which D consults each time it loads. The developer can set the maximum number of string and numeric variables to a value other than the default of 50 and can include Color, Crtspec, and Pagespec commands to define screen colors, screen size, and printer-page size for an application. Developers can customize their full-screen logo by defining it in the CONFIG.D file. D displays the custom logo after it displays its own logo.

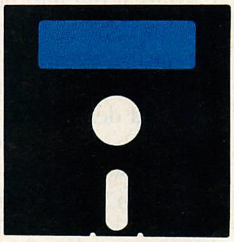
Although some D commands duplicate the function of DOS's CD, Dir, and Erase commands, no shell command exists to exit to DOS, execute a program, and return to D. A Call command runs external .EXE files written with the use of D's *host-language interfaces*, sets of object files that come standard in D and must be linked with external programs written in Microsoft or Lattice C or Microsoft Assembler. The amount of memory the program requires can be specified and parameters can be passed to the external program. The Call command cannot execute programs that do not incorporate the host-language interface. Any attempt to do this causes D to terminate abnormally, which results in a possibly corrupt database.

External programs invoked with Call operate essentially as subroutines to D. They perform functions external to the database and place results in D's dialog history.

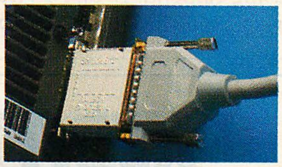
For example, an external program could extract data from a CAD drawing file and place them in a data file that subsequently could be loaded into the D database. A program written in C, incorporating both the D host-language interface and Premier Design Systems' AutoDirect2, could extract attributes or geometric data from an AutoCAD drawing and import them into a D database without leaving the D application. In fact, an application incorporating AutoDirect2 could modify the AutoCAD drawing file without ever having to leave D.

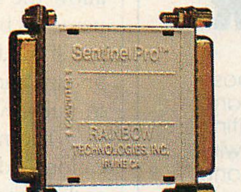
Importing and exporting. The developer imports and exports data using the Load and Unload commands. D can import Data Interchange Format (.DIF),

In 1988, $3.5 billion in micro-computer software will be sold worldwide. During that same time, another



$3.0 billion in sales will be lost to free distribution — better known as software piracy. And right now, Rainbow Technologies' Software Sentinel™ is protecting close to $1.0 billion in software for developers who never wanted to be part of the free software distribution network in the first place. The Software Sentinel hardware key is "execution control" software protection. It ships with the software and simply plugs into the PC's parallel port to be one hundred percent invisible to both user and the software. Users can make as many copies as they want. Make working submasters. Use a hard disk. Virtually anything that can be done with unprotected software. Except start freely distributing that software to other users. The Rainbow family of Software Sentinel products. Selected by the very big to the not-so-big developers of DOS, OS/2 and Xenix software in worldwide markets. To the cool tune of close to a billion dollars. So far.





RAINBOW TECHNOLOGIES

18011-A Mitchell South, Irvine, CA 92714 • (714) 261-0228 • TELEX: 386078 • FAX: (714) 261-0260
Rainbow Technologies, Ltd., Shirley Lodge, 470 London Rd., Slough, Berkshire, SL3 8QY, U.K., Tel: 0753-41512, Fax: 0753-43610

CIRCLE NO. 157 ON READER SERVICE CARD

ASCII fixed-length, ASCII delimited, dBASE (.DBF), and blocked-format files, and can export all but .DBF.

Load and Unload work with record descriptions in personal interfaces. A record description describes an external data file. D breaks records of the external file into fields according to the record description. It then matches the record description with the data-group definition based on field names. If the record description and data-group definition have fields with the same name, data flow from the external file, through the record-description fields, to the data-group fields. Field order and length are significant between external file and record description; field names are significant between record description and data-group definition.

Because record descriptions are not part of the general interface produced when the database definition is compiled, a personal interface generally must be created and compiled before records are imported or exported.

In addition, data can be imported from mainframe and minicomputer file formats such as block files, with no conversion routines necessary.

Maintaining the database. When a record is appended using Append, D updates the index structure in the tree file automatically. When a database is loaded, however, the tree and list files are not updated automatically. A Load operation must be followed by an explicit Construct or Structure command. Construct adds to an existing index structure and Structure creates a new one. If an index structure exists, Structure must be preceded by a Destruct operation to remove the existing index. The Structure command can build the index for an entire database (key fields only) or for a single key field. The Construct command can update the index for the entire database (key fields only), a data group's key fields, or a single key field.

Keys can be added with the Key command, and their uniqueness can be specified. Removing a key requires recompiling the database or erasing the index structure with Destruct and then rebuilding it, one field or data group at a time, omitting keys not required.

RUNNING WITH THE BEST

The *PC Tech Journal* editorial inventory sample application was implemented in D to test the ease of developing applications and to measure their performance (for details, see the August 1985 article referenced earlier). D scores well in both areas. Figure 2 shows the code for benchmarks 3 (documenting and tallying codes from one column), 4 (mass changing of one column), and 5 (extracting selected records to a text file). Only 15 lines of code were required for all three benchmarks.

Defining the database was straightforward, taking only an hour even with a few detours. This involved trying to group the Volume and Issue fields of the Issue data group into a field called Issue, only to find that D does not allow the field name and data-group name to be identical. Further, developers should know up front that data groups can be related by equal values in fields with different names.

The meat of the benchmark application is defining procedures, reports, and screens. After creating the database definition, only four one-line procedures are needed to load the data files and one to create the index structure. The report definitions for the benchmarks required less than one screen of code. A mailing-labels report was printed easily using a single statement after the collection was arranged.

Ad hoc queries are constructed like reports. D performed each query specified for the benchmarks with two statements: a Find or a Look statement to isolate the records of interest and a Print statement to display the records and relevant calculations or statistics. Several query procedures written using the Look command provided an interactive means of obtaining user input with minimum code.

FIGURE 2: *Sample Code*

DOCUMENTING AND TALLYING CODES
```
Clear A
Fill A Author
Reduce A Author State
Sort A by State
Print A State
Type done............
Pause
```

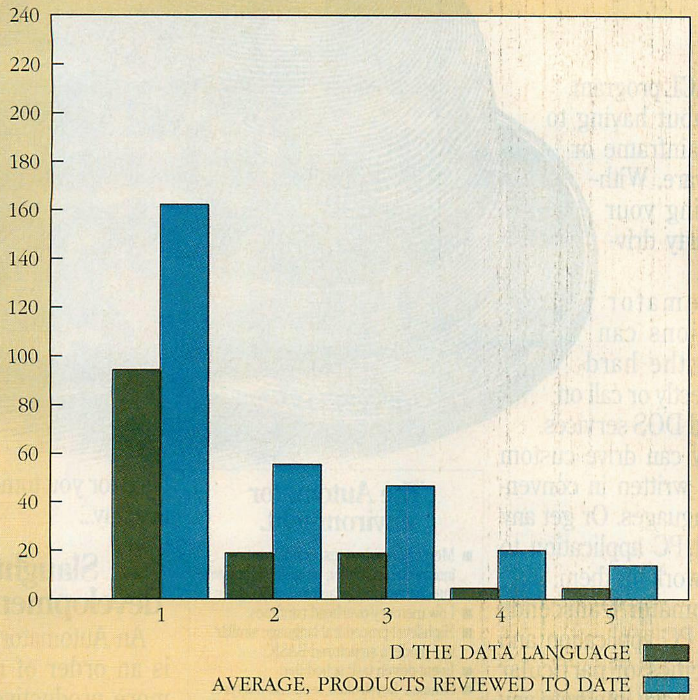
MASS CHANGING OF ONE COLUMN
```
Clear A
Find A Author where State = CO
Change A Author State to CL
Checkpoint
```

EXTRACTING SELECTED RECORDS
```
Clear A
Find A Author where State = CA
Sort A by ZIP
Unload A Author Pctauthor AUTHORX.ASC
```

The D language, used for procedures and queries, is English-like, terse, and based on the real-world method of isolating, sorting, and reporting data. Very few lines of code were needed for the *PC Tech Journal* benchmarks.

FIGURE 3: *Performance Benchmarks*

BENCHMARK TASK	TIME
1. Add 900 records to an empty database table	94
2. Index table on two fields (7 bytes)	19
3. Document and tally codes from one column	19
4. Mass change of one column (28 rows of 900)	4
5. Extract selected records to create a text file	4



All times are in seconds.

All benchmarks were run on an IBM PC/AT (6 MHz) with 640KB memory. The tests were run in an 8MB partition on a CMI 20MB hard disk under DOS 3.0.

D performs better than the average data manager tested to date in all *PC Tech Journal* benchmarks; it is among the top three performers in benchmarks 2 and 4, the top five in benchmark 3, and the top seven in benchmarks 1 and 5.

A query to print overdue articles for an issue proved to be a challenge because there is no date type in D; to do date arithmetic, date strings must be converted to Julian form. Because D includes a conversion function, this was only a slight inconvenience.

Creating a point-and-shoot menu was simpler than creating procedures and report definitions because no code exists. The menu was created in just a few minutes by typing procedure names, placing the cursor in front of each name, pressing F5 to mark lightbar locations, and saving the file.

To meet the *PC Tech Journal* benchmark specification, the application must include a screen that performs data validation against lists and ranges, retrieves data from multiple files (data groups), and calculates virtual fields from actual fields. Developing this screen in D was quick and simple, the only disappointment being that D has no facility for reading the screen text from the external file furnished with the benchmark data files. Again, there is no code for a screen, except for optional special procedures.

Creating the basic screen took about 30 minutes and iterating through special procedures, ranges, and lists to implement the benchmark's editing specifications took approximately another hour. D's power was immediately apparent; creating a screen with access to multiple data groups was transparent. Defining a transfer field required only a valid field name from the database definition; the developer need not indicate the data group.

Not only was development quick in D, but also performance was excellent. Figure 3 reveals that D performed all operations, except import, three to four times faster than average.

COVERING ALL THE ANGLES

D is suitable for a wide range of applications from simple mailing-list management to complex order-entry and inventory control. Mailing-list management hardly even taxes D's capabilities; a five-line menu, a single one-line report definition, and one screen provide facilities to load and backup the database, isolate a set of records, arrange records, and print labels. It is difficult to imagine even a canned mailing-list program being any simpler.

Benchmark results indicate that D also is suited to complex applications. Few data managers allow solutions to the benchmark's specifications that are as compact as the solutions in D. Because the database is limited only in length of records and total file size—a DOS limitation that has been relaxed with the release of DOS 4.0—D can host large, complex applications.

While data-processing power is easy to come by in many data managers, features that improve the developer's productivity, as are found in D, are not so readily available.

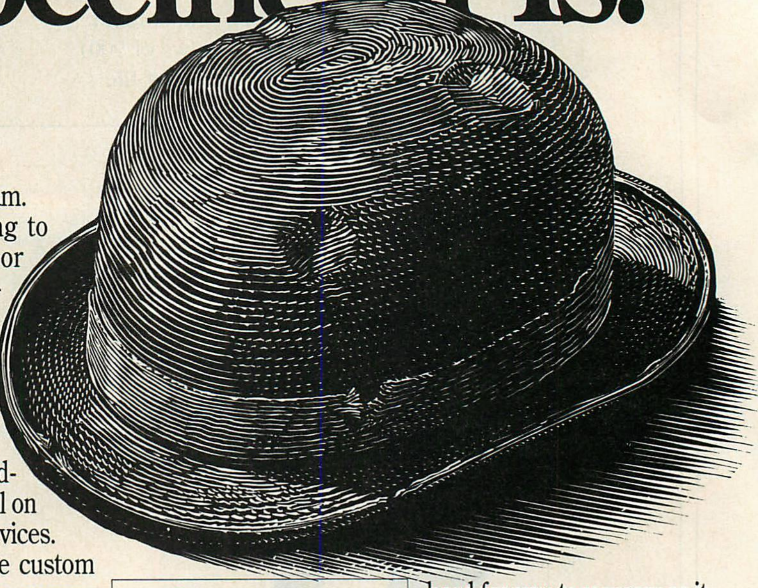
D provides the developer with a highly interactive development environment. Because procedures are interpreted, developers can test them immediately after defining them. Yet, the compactness of the D language—the D version of the benchmark was approximately 20 percent the size of the dBASE III PLUS version—affords applications speed of execution higher than most other products, interpreted or compiled. The highly interactive environment combined with dense code indicates that order-of-magnitude improvements in developer productivity are not only possible, but likely in D.

D is ideal for developers who prefer the control of a procedural language, but need to achieve high productivity. This product's environment is equal to that of the best application generators in terms of enhancing developer productivity.

D is powerful, yet compact, easy to learn, and easy to use. It is currently an excellent choice for small business. When D becomes multiuser with the introduction of the Xenix version in the first quarter of 1989 and the DOS network version in spring of 1989, it will be able to compete even for big-business customers.

Victor E. Wright is a freelance writer and CAD/CAM/CIM consultant in Louisville, Kentucky. He has a degree in industrial management from the University of Cincinnati and is a registered professional engineer.

Whatever happened to emulator-specific APIs?



Linking PCs to mainframes may be old hat. But to do the job properly, you need to build on top of your emulator, creating a custom application that's both bullet-proof and idiot-proof.

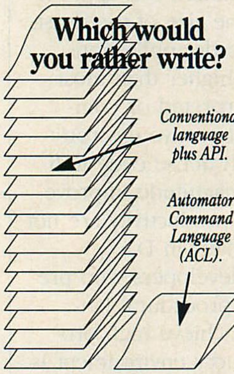
Many developers are still doing it the hard way. Using emulator-specific APIs, conventional languages and months of hard labor.

But instead of months, you can now do it in days, or even—literally—hours.

With Automator, from Direct Technology. A unique, award-winning development system, designed to tackle exactly this kind of problem.

A brilliant new approach.

Automator supports a comprehensive high-level language, ACL, tailored for the intelligent automation of PC-to-mainframe links.



ACL code gives you high-level control of your emulator and DOS applications. All the functionality you need is built-in. Your code is shorter, clearer, easier to write and easier to maintain.

ACL programs run resident, seizing control of the PC and any foreground application whenever necessary.

ACL gives you the kind of power that you usually have to create for yourself. Such as event-handling, task scheduling and concurrency. And the remarkable ability to simulate, under program control, *anything* a human operator can do.

Yet ACL has all the meat you need to actually write your application; from disk file handling and low-level interfacing, to window and menu support.

The universal scripting language.

Pull data off a mainframe, shift it into Lotus and print a graph—all under control of your ACL program.

Without having to adapt mainframe or PC software. Without getting your hands dirty driving APIs.

Automator applications can address the hardware directly or call on BIOS and DOS services.

They can drive custom software written in conventional languages. Or get any standard PC application to do their work for them.

Automator transcends any one PC application and the specifics of particular APIs. So it can integrate any PC applications, whether they have APIs or not.

Giving you a single, generic standard for automation and integration. And high-level code that is easily portable, easy to adapt and easy to maintain.

Not Version One.

With Automator, you're developing your own software, not beta-testing somebody else's.

Automator has been the secret behind successful comms projects at over 800 major corporations worldwide since 1984.

And we're committed to keeping the product up-to-date with advancing technology.

So you can be sure that Automator will be working as hard for you tomorrow as it is today...

The Automator environment.

- Memory-resident text editor, runtime interpreter, compiler, program generator, interactive window and menu designer.
- Low memory-overhead runtimes.
- High-level procedural language similar to a modern structured BASIC.
- Event-driven task scheduler.
- Simulate keystrokes, read and interpret data directly from screen.
- Disk file support from within ACL.
- Built-in windowing and menu support.
- Plus... named procedure calls, program and variable chaining, numeric and string arrays, math operators, PEEK, POKE, IN, OUT... and much more.

Slaughter development time.

An Automator developer is an order of magnitude more productive than the same person working with an API and a language such as C, BASIC or Assembler.

Chances are, you could get your next PC-to-mainframe project to prototype in less time than it takes to read our comprehensive Technical Overview.

For your free copy, clip the coupon or call 1-800-992-9979; in New York City, call (212) 475-2747.

YES! I'd like to see how Automator can slaughter PC-to-mainframe development time. PCT1

☐ Rush me a FREE copy of your Technical Overview.
☐ Call me soon to arrange a demonstration.

Name ....

Title ....

Company ....

Address ....

.... Zip ....

Telephone ....

Return coupon to:
Direct Technology, Inc.
10 East 21st Street, Suite 1204/5,
New York, NY 10010.

AUTOMATOR
From Direct Technology.

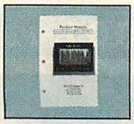
FOR THE NAME OF YOUR NEAREST DEALER CALL 1-800-992-9979.

Automator is compatible with IBM PC, XT, AT and 100% compatibles, and PS/2 model 30, 50, 60, 80.
Most communications products, including the leading 5250, 3270 and VT100 emulators from AST, DCA (IRMA), IBM (except Workstation Program 1.0 and 1.1), Rabbit, IDEA, ITT, CXi, Techland (Blue Lynx), INS and Honeywell VIP.
Most popular networks, including Novell, Gnet, IBM Token-Ring, IBM PC Net, Banyan, 3Com and Torus.
Direct Technology, Inc. 10 East 21st Street, Suite 1204/5, New York, NY 10010. Telephone (212) 475-2747. Facsimile (212) 529-4941. Direct Technology acknowledges all trademarks used in this advertisement.

CIRCLE NO. 124 ON READER SERVICE CARD

PRODUCT WATCH

Reviews and Updates

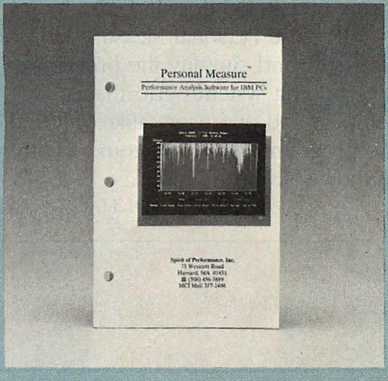


PERSONAL MEASURE 1.0
Spirit of Performance

AUTOMATOR MI 2.2G
Direct Technology Limited

PERSONAL MEASURE 1.0
Spirit of Performance
73 Westcott Road
Harvard, MA 01451
508/456-3889

PRICE: $69.95 (5.25-inch diskette)
$74.95 (3.5-inch diskette)



CIRCLE 337 ON READER SERVICE CARD

Identifying the proper hardware and software components for a system while balancing price and performance is too often a black art. Personal Measure 1.0 from Spirit of Performance augments traditional evaluation tools and analyzes system performance while applications are actually running.

The heart of the package is PMEASURE, a terminate-and-stay-resident (TSR) program that collects and records the time a system spends in each of nine DOS and BIOS interrupt routines (see table 1). The companion program, PMANALYZ, then analyzes and displays a graph of the results.

The developer can analyze an entire file, a specific time slot, or a specific program and can view the summary statistics underlying the graph. Both the summary statistics and the graph can be printed.

To test the package, we analyzed the performance of a text editor and two source-code processors (see photo 1). Time is represented by the horizontal axis and percentage-use, by the vertical axis. Each vertical stripe represents a slice of time, color-coded to show the percentage-use of the five system components: processor, disk, keyboard, printer, and auxiliary port.

In the test, four distinct sections are notable. At 9:14:25, the text editor begins to run. The graph is primarily blue, signifying the system spends most of its time waiting for keyboard input.

When the first source-code program is run from within the text editor (9:14:45), the graph shows 60 percent yellow (disk I/O) and 30 percent green (processor). At 9:16:00, when the editor is stopped and control is returned to DOS, the graph again is primarily blue and waiting for keyboard input. The second source-code program is started at 9:16:10. The analysis indicates mostly processor activity with a small amount of disk I/O.

The analysis of system use provides insights about the application software's performance. Predictably, when the text editor and DOS are run, most of the system's time is spent waiting for keyboard entry. The first program is I/O-bound; more time is spent in reading and writing to the disk than in processing. Conversely, as can be determined from the graph, the second program is processor-bound.

Personal Measure also produces two summary reports of system resource use and disk activity. The first table lists the amount of time (to one one-thousandth of a second) each component was used and the corresponding percentage.

The second table summarizes file reads and writes, directory and non-DOS accesses, and the number of programs loaded. The number of DOS and BIOS calls and the duration of each activity is also totaled. The final bit of information provided is the average BIOS disk access time.

Another useful feature of the summary tables is an "advisory message," which provides possible reasons for performance problems. If the ratio of BIOS to DOS calls for file reads, for example, is greater than 1.5, the message "file fragmentation?" suggests that the user should invest in a file optimizer or reorganizer.

After the program identifies specific problems, the user can turn to the manual for suggestions on improving system performance. Chapter six explains inexpensive software and hardware changes, such as disk caching and the virtues of the NEC V20, and gradually works up to the ultimate 386 solution. Specific products are recommended. Combined with chapter seven, a summary of available benchmark programs, these two chapters make the manual a valuable product in itself.

Because the software monitors the BIOS and DOS interrupt routines, Personal Measure will not measure devices accessed by special drivers or other low-level techniques—including Iomega's Bernoulli Box and Adobe PostScript printer drivers. The package does not attempt to isolate video-terminal I/O because very few programs use the BIOS and DOS video routines.

In addition, programs that expect to have exclusive control of the disk can cause problems because Personal Measure writes to the disk in the background. These programs include disk utilities, formatters, and optimizers, but they are run only occasionally and do not represent a significant problem.

PMEASURE uses less than 10KB of memory while running. Because of this frugal use of RAM, it should not interfere with most programs. If other TSRs have been loaded after Personal Measure, the program remains resident but becomes inactive.

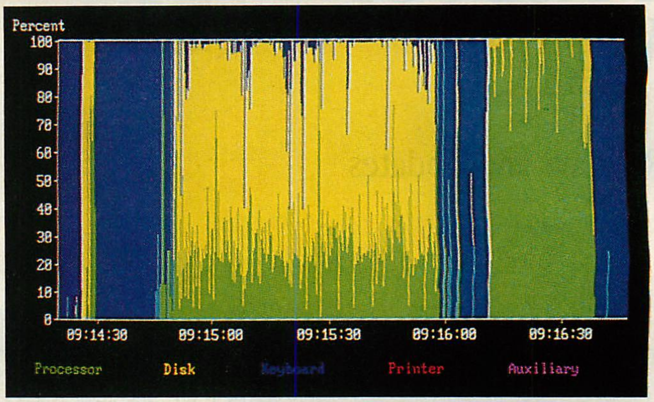
Personal Measure's developers are well aware of the pitfalls and problems inherent in any TSR application. The

TABLE 1: ***Interrupts Monitored***

INT	FUNCTION
13H	BIOS disk/diskette services
14H	BIOS serial-port services
16H	BIOS keyboard-input services
17H	BIOS printer-port services
20H	DOS program terminate
21H	DOS services
27H	DOS program terminate-and-stay-resident (TSR)
33H	Mouse-driver services
5CH	MS-NET transport services

Personal Measure monitors the flow of information through software interrupt vectors to analyze hardware use.

PHOTO 1: ***Performance Analysis***



Analysis shows percentage-use of system components when running a text editor and two source-code processors.

manual contains clear and detailed advice on running the software with other TSRs and lists programs that conflict with Personal Measure.

Anyone evaluating hardware for purchase should consider benchmarking the systems by running real-life programs under Personal Measure. Information centers and retail stores can use the software to help clients make informed decisions about buying hardware and software.

C language programmers often analyze programs using a profiler to measure each function within the program. These developers also can use Personal Measure to evaluate another program's overall performance by examining its performance when prepared with different compilers, window toolboxes, or file-processing routines.

Despite the vagaries of the DOS environment, the developers at Spirit of Performance have successfully implemented a performance-measurement technique that works consistently on most programs. The package has a single, clearly stated function that it performs in a straightforward manner.

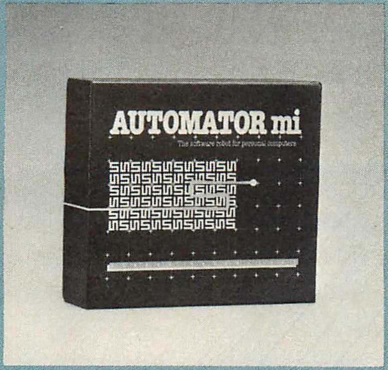
Because Personal Measure helps to demystify and simplify the process of determining system performance, developers and end users alike will find the program to be a welcome addition to their software libraries.

—JOHN HAGSTRAND

AUTOMATOR MI 2.2G

Direct Technology Limited
10 East 21st Street
Suite 1204
New York, NY 10010
212/475-2747

PRICE: $1,995



CIRCLE 339 ON READER SERVICE CARD

Automator mi (mainframe interface) 2.2G from Direct Technology Limited is a rare find. The company says Automator operates like a software robot that monitors a PC and performs actions based on what it senses—a metaphor that can be extended to any reasonably powerful programming language. What makes Automator worthy of a close look are the capabilities it provides for the little effort required.

Automator is a terminate-and-stay-resident (TSR) tool that monitors the screen, keystrokes, and system clock. It is designed for the automation of micro-mainframe communications, but can simplify other tasks as well.

As a programming tool, Automator has the capability to manipulate a PC at its most fundamental levels. Automator command-language (ACL) scripts produce TSRs that can do everything from monitoring CPU registers to painting menus. The developer can use the package to reduce the complex process of logging onto a mainframe (using arcane utility and data-access commands) to a few simple PC commands.

ACL contains most standard programming-language functions, such as subroutines, Boolean and arithmetic operations, labels, and string manipulation. Specialized operators enable applications to monitor events and to read from and write to video buffers.

ACL can simulate keystrokes, intercept keystrokes before they reach an application, paint overlapping windows on the screen, and write and read sequential and indexed files. ACL also can peek and poke data into memory, read and write data to I/O ports, set up registers, and issue interrupts.

Using Automator, a developer easily can create TSRs, such as those illustrated in figure 1—a sample of ACL code that is an adaptation of a Direct Technology Limited application. The program displays the current date and time in the upper-right-hand corner of the screen, determines if the display is color or monochrome, and sets the mode accordingly. The location of the display is set with the window commands. Automator uses the window construct to define where its application either will read from or write to the screen display.

Writing ACL scripts is quick, and Direct Technology's development environment makes generation of ACL scripts quite effortless. The system is an integrated TSR set of functions for creating, editing, and running the scripts, which also includes a code generator for building small, complete applications or for building skeletons for more complex situations.

The generator operates in the same fashion as the applications: it monitors what the operator does and creates ACL code fragments that mimic the operator's actions. The developer invokes the generator from the main menu, which is accessed with a hot-key combination. The program collects the fragments in a buffer, which the developer can embellish with the included editor. The development kit includes the editor and an interpreter; together, they help speed up the development cycle. A stand-alone compiler (included in the development package) generates pseudocode that is executed by an optional runtime interpreter.

PC configurations can be complex, particularly given the many possible combinations of hardware and software. To be resilient, software packages interacting with a mainframe through a terminal emulator must handle unusual errors, such as slow or incomplete mainframe response or loss of network connection. Automator checks for prolonged waits or invokes procedures to handle exceptional situations.

Most applications coexist peacefully with Automator, including Lotus 1-2-3, Borland's SideKick, and Microsoft Word. According to Direct Technology, Automator is compatible with Novell's NetWare, 3Com's 3+, and IBM's Token-Ring networks, but will not work with Nestar and Apricot networks.

FIGURE 1: ***Sample Code***

```
;
; AUTOMATOR mi - LEarn program ;
;

whenever '<Ctrl Esc>' hit stop ; stop with
                               ;Ctrl-Esc
if _mode = 7 ; if display is monochrome
    _dispcol = {x70 ; set to reverse video
else
   _dispcol = {x84 ;set to flashing red text
endif
repeat
    window 0 0 13 66 24 ;time-display spot
    h = _hours  ;system hour variable
    mins$ = ""  ;string variable
    if _mins < 10
        mins$ = "0"
   endif
    mins$ + _mins ;auto conversion of string
                  ;to integer for addition
                  ;to system minute value
    display " Time " h ":" mins$
    window 0 0 20 44 24 ;spot for date display
    display "Date  " _day$ " " _day "-"
            _month$ "-" _year
    wait 2 secs
until 1 = 2     ;forever
```

This ACL script determines the type of display in use and exhibits the current date and time on the screen.

An application designed to retrieve E-mail from a mainframe was used to test some of Automator's capabilities. The application works well with a PC connected to an IBM mainframe using DCA's IRMA board and software. The application also performs well using a US Robotics modem and Relay Gold software, but did not work when using an IBM emulator board with IBM's 3270 Workstation Program. Direct Technology says that Automator will not automate an IBM 3270 Workstation Program emulation session, although it will work within one DOS session running under the workstation application.

Although Automator is an impressive package, it is costly in both PC memory and price. The complete development package, including its online help, uses almost 276KB. Automator's runtime interpreter consumes another 84KB of memory before the pseudocode is included. A single development package costs $1,995 (which includes one runtime set). Separate runtime sets are $200.

Automator's price is not unreasonable for corporations or systems integrators. Because Automator mi can save a developer time developing classes of applications that are notoriously tricky, for many situations it is indeed a bargain.

—PAUL FIRGENS

SIMPLE SOPHISTICATION FOR THE PC WORLD.

The V-series Smartmodem 9600.

The Hayes V-series™ Smartmodem 9600™ brought a technical sophistication to the high-speed dial-up world that had rarely been seen before.

With error-control for point-to-point communications, it automatically detects errors and then retransmits the data correctly.

With adaptive data compression, it has the capability to double throughput to up to 19,200 bps.

And with Hayes AutoSync, it provides both synchronous and asynchronous communications without the extra expense of a synchronous adapter card.

The Hayes V-series Smartmodem 9600 has a sophistication that not only sets it apart from other PC modems, but above them as well.

SOPHISTICATED SIMPLICITY FOR THE MAINFRAME WORLD.

Introducing the Smartmodem 9600 for V.32 communications.

The new Hayes Smartmodem 9600™ brings the simplicity of the PC world to the complex world of wide and local area networks.

For host to host, host to LAN, and data processing centers to remote controllers, the new Hayes V.32 modem gives full-duplex, synchronous and asynchronous communications at 4800 bps and 9600 bps using dial-up or leased-line circuits.

The modem also offers features like automatic leased-line restoration, automatic dial backup and forward error correction to ensure reliable data transmission. An intelligent front panel with an LCD readout puts this array of features at your fingertips. While behind the modem, the Hayes Cable Management System lets you easily install it flush against any wall or cabinet to save space. The Hayes Smartmodem 9600. We're not just entering the mainframe world, we're improving it.

Hayes®

For your nearest Hayes Advanced Systems Dealer, call **800-635-1225.** Hayes Microcomputer Products, Inc., P.O. Box 105203, Atlanta, GA 30348.

CIRCLE NO. 180 ON READER SERVICE CARD

1024X768 IBM COMPATIBLE VGA VIDEO CARD

VGAWONDER™

THE SECOND WONDER

1024X768

Plus: VGA, EGA, CGA, MDA & Hercules on Standard MULTISYNC monitors

Yes! the Phenomenal VGAWONDER does it all at IBM Hardware level compatibility with automatic monitor detect and no dip switches or jumpers to set. A HOT CARD for the 286/386 POWER USER with AUTO BUS DETECT for 8 or 16 bit slots, running with a 16 bit datapath with 1:1 interleave. 100% IBM HARDWARE LEVEL COMPATIBILITY means that ALL software and operating systems designed for IBM systems will run. 1024 WILL BE DISPLAYED IN COLOR ON STANDARD MULTI-FREQUENCY MONITORS. Although these features are expected from ATI they are NOT AVAILABLE from other manufacturers. EGA on all monitors including RGB & TTL means a perfect card for networks. MOUSE ON BOARD! ATI built the mouse right in to this one, why not? No extra charge! VGAWONDER is available in both 512K and 256K Versions. Ask your dealer for more information.

MOUSE INCLUDED

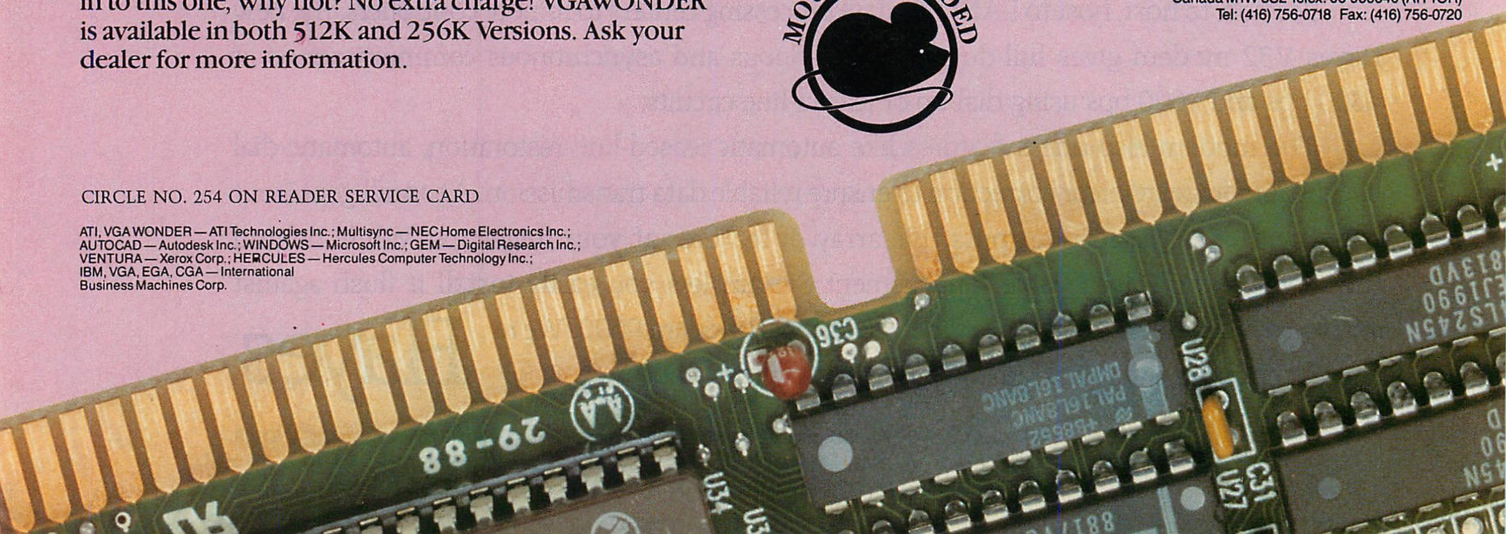
ATI® TECHNOLOGIES INC.

Technology you can Trust.

ATI Technologies Inc.
3761 Victoria Park Ave., Scarborough, Ontario
Canada M1W 3S2 Telex: 06-966640 (ATI TOR)
Tel: (416) 756-0718 Fax: (416) 756-0720

CIRCLE NO. 254 ON READER SERVICE CARD

ATI, VGA WONDER — ATI Technologies Inc.; Multisync — NEC Home Electronics Inc.; AUTOCAD — Autodesk Inc.; WINDOWS — Microsoft Inc.; GEM — Digital Research Inc.; VENTURA — Xerox Corp.; HERCULES — Hercules Computer Technology Inc.; IBM, VGA, EGA, CGA — International Business Machines Corp.



TECH NOTEBOOK

A forum for sharing solutions to technical problems

1 SHARED MEMORY

2 OS/2 QUEUES

Even at its best, documentation for a system as complex as OS/2 is not light reading. The problem is inherent in the documentation itself. It is meant to *state*, not *explain*; to give the *what*, not the *why*. In many cases, explanation is more crucial than exposition, and many system services are incomprehensible without some illuminating insight beyond a mere statement of the facts. The situation is aggravated if the reference materials are incomplete, inadequate, misleading, or just plain wrong.

Some of the worst transgressions in the OS/2 reference manuals occur in the documentation of queues, a mechanism for interprocess communications. The two culprits are IBM's *OS/2 Technical Reference* and the *Programmer's Reference* volume in Microsoft's OS/2 Programmer's Toolkit. No one could successfully implement queues from the information given in these manuals. The most significant missing piece of information is that queues are primarily a means of passing pointers to shared memory. The second item this month presents explanations and examples of queue usage; first, however, we need to fill in some gaps in the documentation of shared memory.

1 SHARING MEMORY IN OS/2

A multitasking operating system such as OS/2 has two conflicting needs. The first is memory protection to isolate one task from another. The second is a mechanism of interprocess communications to connect one task to another.

The most efficient means of communication is shared memory: one process writes data into memory that another process can read. Each process that has access to the shared memory needs a selector describing the memory in its local descriptor table (LDT). When a process deallocates a shared memory segment, the segment is removed from the address space of that process but continues to exist in the address spaces of any other processes that have selectors for it. The segment is destroyed only when the last sharing process deallocates it.

OS/2 provides three methods for sharing memory: naming, giving, and getting. Named memory is conceptually the simplest method and is adequately described in both IBM's and Microsoft's documentation. For the other two, the documentation omits some important details. Further, the manuals provide no clue on how to choose among the three methods.

Using named memory is similar to the writing of a file by one process to be read by another. The difference is that a file is permanent and exists even in the absence of any processes, while shared memory exists only as long as it is owned by at least one process.

A process creates a named memory segment by calling DosAllocShrSeg. One of the arguments to the call is an ASCIIZ string giving the segment's name; it has the same form as a file name, but must begin with the reserved name \SHAREMEM\. The system returns a selector for the segment in the address space of the calling process. Any process may gain access to this same segment by calling DosGetShrSeg, passing the segment's name; the system returns a selector for the segment in the LDT of that process. Each process then accesses the named segment with a different selector.

The second method, giving memory, is totally under the control of the process that initially allocates the shared memory. A process allocates such a segment by calling DosAllocSeg with the SEG_GIVABLE flag set. The system returns a selector for this segment in the caller's LDT. The creating process may give access to the segment to another process by calling DosGiveSeg, passing this selector and the ID of the receiving process. OS/2 maps the segment into the LDT of the receiving process, but returns the receiver's selector to the donor. This crucial detail is inadequately stressed in the documentation; the returned selector is for the *receiver's* address space, and it is invalid in the donor's address space. The donor process is responsible for passing this value to the receiver in some way—by way of a queue, for example. As with named memory, each process refers to the segment with a different selector.

In the third method, getting memory, each process refers to the shared segment with the same selector. The sharing mechanism, not explained in the documentation, is similar to the way processes share access to code and data in dynamic link libraries (DLLs).

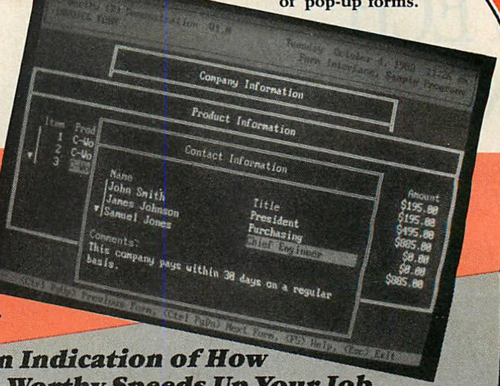
The creating process allocates the shared segment by calling DosAllocSeg with the SEG_GETTABLE flag set. OS/2 maps the newly created segment into the *disjoint LDT space*, which is a set of selectors reserved for shared memory and DLL segments in all processes (see "OS/2's Dynamic Link," Mary DeWolf and Ted Mirecki, September 1988, p. 100). For example, if the newly created segment uses selector n in the LDT of calling process, then OS/2 reserves selector n in every existing LDT and every LDT created subsequently. The memory, however, is not automatically allocated to each process—the reserved selectors are marked unused but unavailable.

Upon return from DosAllocSeg, the selector is valid only for the process that allocates the memory, meaning that initially only this one process owns the segment. To allow sharing, the owner passes the segment's selector to some other process; this process then calls DosGetSeg to map the shared segment into the same selector in its own address space.

To the Professional Programmer:

We Want to Make Your Life Easier.
A Whole Lot Easier.
We want to Send You the One "Human Interface" that Keeps Your Application Program in Control – No Matter What the User Does... or What Mistakes He Makes.

An example of a scrollable Repeated Groupfield plus a word wrapping text field – part of a series of pop-up forms.



C-WORTHY®

The Interface Library with More than 400 Functions Replacing 7 Major Library Categories with One Superfast, Powerful, Integrated, "Bullet-Proof" Software Development Tool

- Windows
- Screens
- Menus
- Error Handling
- Context-Sensitive Help
- Keyboard Handling
- DOS Interface
- Word Wrap Text Editor
- Data Input Forms (optional)

P.S.: No royalties to pay.

A Few Tech Specs

- Compilers: Microsoft 3.0+, Quick, Turbo, Lattice. All models.
- Menus: Fully support pop-up, Lotus style, MS Windows style (pull-down), pull-up.
- Errors: DOS, program, and user.
- DOS Interface: 62 functions.
- Help: System and context sensitive.
- Screens: Screen display, color palettes, save, restore, scroll, more.
- Editing: String and word wrap test.
- Form Interface Library: 118 functions.
- Compatible with MS Windows.
- OS/2 version available.
- Machines: Autodetect for MDA, CGA, EGA, VGA.

C-Worthy **will replace a whole batch of separate library categories... and handle every one of those categories better, faster, and easier. Worth a try at *our* risk? Why not?**

The C-Worthy Interface Library – Direct Price

Object only	$195
Form Interface Library Add-On	$100
Object with Forms	$295
Object with Forms and Library Source	$495

(Please specify compiler and version when ordering)

Price includes 700+ page Reference Guide and 250 page User's Guide with Tutorial and "Getting Started" sections.

Your absolute money-back protection:
The Most Extraordinary Guarantee in the History of Progamming Software!

C-Worthy Interface Library will handle every aspect of your program's human interface. When you buy C-Worthy, use it for two full months before deciding whether you want to keep it. If you return it for any reason, Solution Systems will refund 100% of every cent you paid. What could be fairer, or stronger proof, than that?

An Indication of How C-Worthy Speeds Up Your Job

Hugh Cruickshank, Project Leader Pinetree Software, Inc. (developer of "Maximizer"):
"With C-Worthy we did a year-and-a-half worth of work in 4 months."

Kevin Kingdon, Senior Software Engineer, Novell, Inc.:
"Our software engineering department does *not* use another human interface library, and that says a lot. C-Worthy is a *higher level approach* to human interface than other C libraries."

Clarke Stevens, CAM Software Research Center, Brigham Young University:
"C-Worthy is a great investment. It saves me a lot of time."

To order the *C-Worthy* Interface Library call toll-free:
1-800-852-8523 Ext. 302
or mail coupon

Solution Systems™
541 Main St.
S. Weymouth, MA 02190 PCTJ 12/88

☐ **OKAY, I'll try *C-Worthy* at your risk. If for any reason I decide to return it within two months, I'll send it back for a 100% refund, no questions asked.**

My name ______
Company name if any ______
Street address ______
City ______ State ______ Zip ______
Phone ______
☐ Object only, $195
☐ Form Interface Library Add-On, $100
☐ Object with Forms, $295
☐ Object with Forms and Library Source, $495
(Plus $4 Shipping/Handling)

Specify compiler: ______
Specify version: ______
Disk size: ☐ 5¼" ☐ 3½"
Indicate payment method:
☐ Check for $ ______ enclosed.
☐ Bill to ☐ Visa ☐ MasterCard ☐ American Express
Card No. ______ Expires ______
Signature ______

C-WORTHY by **Solution Systems™**
541 Main Street
South Weymouth, Massachusetts 02190
Telefax: 617-337-7719

Each method has characteristics that make it useful in a particular set of circumstances. Named memory requires no communication between the sharing processes; the process that initially creates the segment need not be aware of the processes that subsequently use it. Giving and getting memory both require a mechanism for passing the shared segment selectors; these sharing methods are typically used between related processes (child and parent processes, for example), or those that establish some other means of communication, such as a queue or named memory.

Sharing memory by giving is the only method that guarantees receipt of the shared segment by the target process. When DosGiveSeg returns, the receiving process owns the given segment. After passing this segment's selector to the target, the donor process can deallocate the segment or even terminate without affecting the receiver's ability to access the segment.

With a named or gettable segment, on the other hand, the creating process has no control over when (or even if) another process gains access to it. The creator must therefore maintain ownership of the segment to ensure that it remains in existence. This approach is useful for a server process that runs continuously, collecting messages from client processes.

Shared memory is an essential part of the mechanism of communicating via interprocess queues. The following item demonstrates the use of each of the three memory-sharing methods for this purpose.

2 COMMUNICATING VIA QUEUES

Queues are a mechanism for sending data from one OS/2 process to another (see "At the Core: An API Comparison," Robert B. Morris and William E. Brooks, this issue, p. 62). The process of creating a queue (with the DosCreatQueue call) gives it a name similar to a file name, beginning with the reserved name \QUEUES\. The creator also specifies the order in which items are added to the queue: first-in, first-out (FIFO); last-in, first-out (LIFO); or by priority. Any process that knows the name can open the queue and write to it; only the creating process (the owner) can read the queue and delete it. Each process that opens the queue is notified of the owner's process ID.

The documentation of queue operations is not only vague and incomplete, but downright incorrect. For DosWriteQueue, the Microsoft *Programmer's Reference* states, "The function copies the element pointed to by the *pbBuf* parameter to the queue . . ." Don't believe it; DosWriteQueue does no such thing.

Elements written to OS/2 queues consist of three components: two words and a double-word. The DosWriteQueue function receives the three components by value and inserts them into the queue without examining them. In particular, OS/2 does not interpret any of the values as pointers, nor does it attempt to access any data to which they may be pointing. The documentation states that the first word is an element identifier (IBM calls it a *request ID*, Microsoft calls it an *event code*), the second is the length of the data being passed, and the double-word is a pointer to the data. OS/2, however, does not enforce this usage, so the values can be anything that is meaningful to the receiving process.

Under most circumstances, the limited information capacity of queue elements makes them practical only for passing pointers to data written by one

process and read by another. For both processes to access this data, they must be written in shared memory, using any of the three sharing methods described in the first item. In addition, special consideration is required to ensure that a pointer constructed in one process's address space can be used in another's space.

To use named shared memory, the queue owner creates a named segment when it creates the queue; a writing process opens both the queue and the shared memory segment by name. For each message, the writer suballocates a region in the named segment, writes the data there, and puts a pointer to it in the queue element. When the queue owner reads the element, it discards the segment portion of the pointer (it is valid only in the writer's address space) and uses the passed offset in conjunction with its own selector for the named segment. After processing the data, the owner deallocates the region containing the passed data, making the memory available for suballocation by another process.

Passing queue data through named memory is useful when many short messages abound; suballocating regions out of one large segment is more efficient than separately allocating a small segment for each message. The limitation is that the total size of all data pointed to by queue elements is limited to 64KB; the queue owner must process the messages quickly to ensure an adequate amount of free space in the named segment.

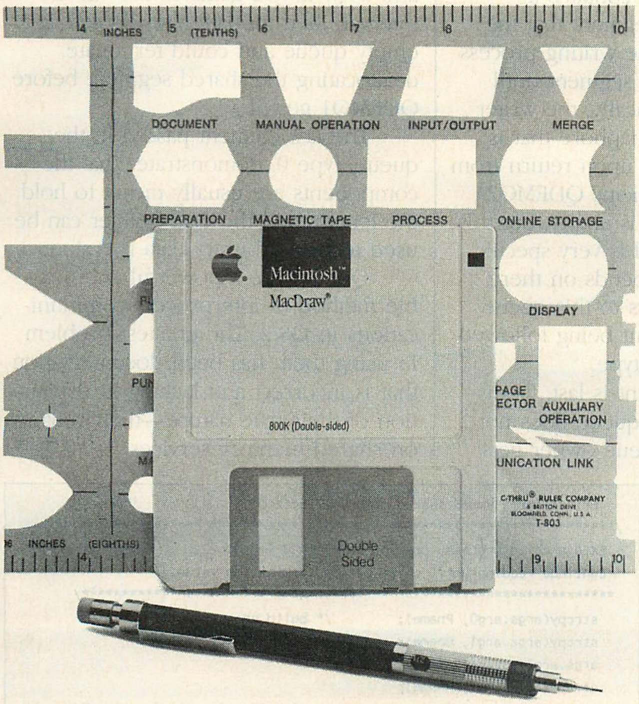
In the second method, giving memory, the writing process allocates a segment large enough to hold the data to be passed to the queue owner, then gives the segment to the owner process. (When the writer opens the queue, it receives the owner's process ID.) The writer uses its own selector for the shared memory to write the data, but constructs a pointer to this data with the selector in the owner's address space, as returned by the DosGiveSeg call. The reading process can use this pointer directly, with no conversion. This is the preferred method when messages are large and need to exist independently of the sender.

In the third method, the writing process allocates a segment with the SEG_GETTABLE attribute and uses its own selector for this segment for both writing the data and constructing the pointer passed in the queue element. The receiving process calls DosGetSeg with the selector portion of the pointer to map the segment into the same selector in its own address space.

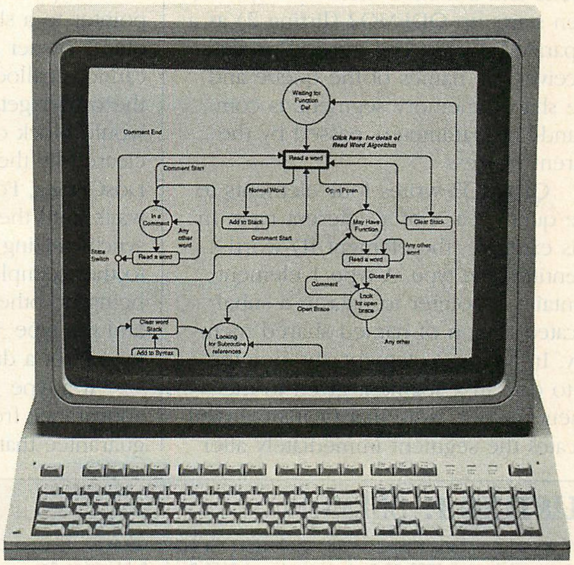
On the surface, giving and getting memory for queue elements appear equivalent. In the former, the writing process calls DosGiveSeg to create a pointer valid in the reader's address space; the reading process uses the pointer directly. In the latter, the writing process passes a pointer valid in its own address space; the reader calls DosGetSeg to validate the same pointer in *its* address space. Each method, however, requires very different means of interprocess synchronization.

After giving memory to the owner, the writing process immediately deallocates the memory containing the passed data (this segment is already owned by the receiving process). If the writing process terminates before its data are received, the data continue to exist in the owner's address space. By contrast, when the owner gets memory, the writer must maintain ownership of the memory until the owner receives it. The owner must therefore incorporate a mechanism for notifying the writer of receipt of the shared memory. This makes getting memory less desirable than giving it.

If you just need a drawing tool, use one of these



If you need a design tool, use Design/2.0™



A lot of tools can help you draw, but how many help you design and keep track of complex system models?

Design/2.0 is the powerful text and graphics processor that allows you to quickly create and edit flow charts, system models, technical documentation, organizational charts, computer programs, communication networks, business plans, and more. And Design/2.0 does it faster and more efficiently than any other tool!

End time consuming redrawing

Design/2.0 understands when objects in a diagram are related. Once you connect two objects, they stay connected. Move or resize an object and its connections are automatically redrawn. Design/2.0 also redraws all associated objects and text.

Use Design/2.0's flow chart symbols or create or import your own. Create and edit text anywhere in your diagram. Even do word searches. Associate text with objects or connectors. Or create hypertext links across pages.

The big picture is easy to see

As your model evolves, move detail to subpages. Build diagrams containing hundreds of hierarchically linked pages. Design/2.0 maintains all relationships and displays the hierarchy no matter how complex your model becomes.

Why not spend your time designing, rather than drawing?

Yes, I want to spend my time designing! Please send me more information on Design/2.0.

Name ______

Company ______

Address ______

City ______ State ______ Zip ______

Mail to: Meta Software, 150 CambridgePark Drive, Cambridge, MA 02140.

Or call 1-800-227-4106. Inside Massachusetts 617-576-6920

Meta Software

PCT 12/88

Available for the IBM® PC-AT, PS/2 & close compatibles. (Requires Windows™) $350
Available for the Macintosh™ Plus, SE, II. $250. Color available for the Mac II.

IBM PC-AT and PS/2 are registered trademarks of International Business Machines Corporation. Windows is a trademark of Microsoft Corporation. Macintosh is a trademark of Apple Computer, Inc. Design/2.0 is a trademark of Meta Software Corporation.

CIRCLE NO. 133 ON READER SERVICE CARD

A pair of C programs printed here demonstrates the various ways of using queues. QDEMO1 (listing 1) creates a queue and a named memory segment, then starts up QDEMO2 (listing 2) as a separate process. This second process receives the names of the queue and the shared memory segment as command-line parameters passed by the parent process.

QDEMO2 writes four elements to the queue, each of a different type. In this example, the element ID word identifies the type. A type 1 element contains a pointer to data in a suballocated region of named shared memory. In a type 2 element, the pointer is to data in a segment given to the queue owner. Note that QDEMO2 deallocates the segment immediately after writing a type 2 element, without waiting for QDEMO1 to read the element from the queue.

A type 3 element contains a pointer to a shared segment that the queue owner gets. The writing process cannot deallocate this segment until the owner gets it; typically, the writer would block on a semaphore that is cleared by the owner upon return from DosGetSeg. For simplicity, QDEMO2 waits until the queue is empty, but this synchronizing method is very specific to the example; it depends on there being no other writers to this queue and the type 3 element being followed by one of a different type.

If a type 3 element is last, its disappearance from the queue does not guarantee that the queue owner gets the associated shared segment. If QDEMO1 is interrupted between the calls to DosReadQueue and DosGetSeg, and if QDEMO2 gains control in the interval, then the latter would see an empty queue and could terminate, deallocating the shared segment before QDEMO1 gets it.

The last element passed in the queue, type 0, demonstrates that the components are usually meant to hold the length, and the data pointer can be used to pass arbitrary data by value.

Queues are a powerful and flexible method of interprocess communications in OS/2. The greatest problem in using them has been documentation that is incorrect and lacking in explanation of their true nature—their reliance on shared memory services.

LISTING 1: QDEMO1.C

```
/* QDEMO1.C - Parent program for demonstrating Queue usage.
 * Copyright (c) 1988 PC Tech Journal and Ziff-Davis Publishing Co.
 * Written by Ted Mirecki
 */
#define INCL_DOS          /* include all DOS* function prototypes */
#include  <os2.h>
                          /* Q item type codes:                   */
#define I_NUM 0           /*    long int, not pointer             */
#define I_NAME 1          /*    ptr to named shared mem           */
#define I_OWN 2           /*    ptr to owner's memory (given)     */
#define I_DISJ 3          /*    ptr to disjoint memory (gettable) */

char fmt[20];         /* Format string for progress messages */
char Pname[] = "QDEMO2.EXE";
char Qname[] = "\\QUEUES\\TEST00.Q";
char Mname[] = "\\SHAREMEM\\TEST00.SEG";
struct {
   char arg0[sizeof(Pname)];
   char arg1[sizeof(Qname)];
   char arg2[sizeof(Mname)];
} args;

SEL   NamedSeg;             /* selector of named shared memory */
HQUEUE Qhan;                /* handle of queue       */
int   err;
PID   progid;               /* process ID of this program */

main()
{
   LINFOSEG far *local;        /* ptr to Local Info seg */
   SEL   Gseg, Lseg;           /* selectors for info segs*/
   char far *outmsg;
   char  missing[65];          /* returned values from DosExecPgm */
   RESULTCODES results;
/*****************************************************************
   INITIALIZE:  create the queue, allocate named shared memory
 *****************************************************************/
   DosGetInfoSeg(&Gseg, &Lseg);   /* get info segments */
   local = MAKEP(Lseg, 0);
   progid = local->pidCurrent;
   sprintf(fmt, "Process %d: %%s\n", progid);   /* create format string */
   printf(fmt, "Parent process started");
                               /* Create the Q */
   err = DosCreateQueue(&Qhan, 2, Qname);
   if (err) errmsg ("Create Q", err);      /* won't return */
   printf(fmt, "Queue created");
                               /* Allocate shared memory */
   #define NamedLen 8000
   err = DosAllocShrSeg(NamedLen, Mname, &NamedSeg);
   if (err) errmsg("Alloc Share", err);
   err = DosSubSet(NamedSeg, 1, NamedLen);
   if (err) errmsg("Sub Set", err);
   printf(fmt, "Named shared memory allocated");
/*****************************************************************
   EXEC a child process, passing names of Q & memory.
   Continue reading until child passes termination value.
 *****************************************************************/
   strcpy(args.arg0, Pname);        /* build arg string */
   strcpy(args.arg1, Qname);
   args.arg1[sizeof(Qname)-1] = ' ';
   strcpy(args.arg2, Mname);
   err = DosExecPgm(missing,          /* buffer for missing name */
                    sizeof(missing),  /* its length */
                    EXEC_ASYNC,       /* don't wait for child */
                    &args, 0L,        /* args, same environment */
                    &results,         /* place for returned values */
                    Pname);           /* name of program */
   if (err) errmsg("Exec", err);
   while (Qread()) {
      printf(fmt, "Press return to continue\n");
      fgetchar();
   }
   DosCloseQueue(Qhan);
   printf(fmt, "Good-bye from parent process");
}
/*****************************************************************
   QREAD(): Read from Q, with wait; determine message type, display
 *****************************************************************/
Qread()
{
   struct {
      PID    writer;          /* ID of writing process */
      int    type;            /* event ID = type of ptr to Q item: */
   } Qid;
   union {
      char far *msg;          /* pointer to received message */
      long     value;         /* value if item not a pointer */
   } Qitem;
   SEL   msgseg;              /* segment of message */
   USHORT msgoff;             /* offset of message  */
   char far *Qtext;           /* pointer to message text */
   int   msglen;              /* other element components */
   char  msgprio;
   err = DosReadQueue(Qhan, &Qid,
                  &msglen, &Qitem,
                  0,           /* read item at head of Q */
                  DCWW_WAIT,   /* wait if Q empty        */
                  &msgprio,    /* priority of item */
                  0L);         /* no semaphore ID  */
   if (err) errmsg("Read Q", err);
   printf(fmt, "Read Q item:");
   printf("    from Process %d\n", Qid.writer);
   printf("    pointer type %d\n", Qid.type);
   printf("    priority     %d\n", (int) msgprio);
   printf("    length       %d\n", msglen);
   switch (Qid.type) {
```

```
        case I_NUM: {
            printf("    Long value    %ld\n", Qitem.value);
            return (0);
        }
        case I_NAME: {
            msgoff = OFFSETOF(Qitem.msg);
            Qtext = MAKEP(NamedSeg, msgoff);
            printf("    Message text:\n%s\n\n",  Qtext);
            err = DosSubFree(NamedSeg, msgoff, msglen);
            if (err) errmsg("Sub Free", err);
            break;
        }
        case I_OWN: {
            printf("    Message text:\n%s\n\n",  Qitem.msg);
            DosFreeSeg(SELECTOROF(Qitem.msg) );
            break;
        }
        case I_DISJ: {
            msgseg = SELECTOROF(Qitem.msg);
            err = DosGetSeg(msgseg);
            if (err) errmsg("Get Seg", err);
            printf("    Message text:\n%s\n\n",  Qitem.msg);
            DosFreeSeg(msgseg);
        }
    }                              /* end of switch */
    return (Qid.type);
}
/*********************************************************************
    ERRMSG: Print error message, terminate process
 *********************************************************************/
errmsg( char *func, USHORT err)
{
    printf("Process %d: Error %d in %s", progid, err, func);
    DosCloseQueue(Qhan);
    exit(1);
}
```

LISTING 2: QDEMO2.C

```
/* QDEMO2.C - Child program for demonstrating Queue usage.
 * Copyright (c) 1988 PC Tech Journal and Ziff-Davis Publishing Co.
 * Written by Ted Mirecki
 */
#define INCL_DOS        /* include all DOS* function prototypes */
#include  <os2.h>
                        /* Q item type codes:                    */
#define I_NUM 0         /*    long int, not pointer              */
#define I_NAME 1        /*    ptr to named shared mem            */
#define I_OWN 2         /*    ptr to owner's memory (given)      */
#define I_DISJ 3        /*    ptr to disjoint memory (gettable)  */

char text1[] = "This message written into a named shared segment";
char text2[] = "This message written into owner's segment";
char text3[] = "This message written into disjoint address space";
char fmt[20];         /* Format string for progress messages */

SEL   SharedSeg;               /* selector for shared memory */
HQUEUE Qhan;                   /* handle of queue        */
int   err;
PID   progid;                  /* process ID of this program */

main(int argc, char **argv)
{
    LINFOSEG far *local;        /* ptr to Local Info seg */
    SEL   Gseg, Lseg;           /* selectors for info segs*/
    PID   owner;                /* process ID of Q owner */
    USHORT offs;                /* offset portion of far pointer */
    USHORT items;               /* count of items in Q */
    char far *outmsg;           /* pointer to Q message */
/*********************************************************************
    INITIALIZE:  open the Q.
 *********************************************************************/
    DosGetInfoSeg(&Gseg, &Lseg);    /* get info segments */
    local = MAKEP(Lseg, 0);
    progid = local->pidCurrent;
    sprintf(fmt, "Process %d: %%s\n", progid);    /* create format string */
    printf(fmt, "Child process started; Args: (Q, Memory)");
    if (argc < 3)  {
        printf(fmt, "Too few parameters");
        exit(1);
```

```
    } else {
        printf(fmt, argv[1]);
        printf(fmt, argv[2]);
    }
    err = DosOpenQueue(&owner, &Qhan, argv[1]);
    if (err) errmsg("Open Q", err);
    printf(fmt, "Press return to continue\n");
    fgetchar();
/****************************************************************
    WRITE Q ITEM IN NAMED MEMORY
****************************************************************/
    err = DosGetShrSeg(argv[2], &SharedSeg);
    if (err) errmsg("Get Shared Memory", err);
    err = DosSubAlloc(SharedSeg, &offs, sizeof(text1));
    if (err) errmsg("Sub Alloc", err);
    outmsg = MAKEP(SharedSeg, offs);
    strcpy(outmsg, text1);
    err = DosWriteQueue(Qhan, I_NAME,       /* msg in named memory */
                        sizeof(text1),      /* msg length */
                        outmsg, 11);        /* address & priority  */
    if (err) errmsg("Write Q, Named memory", err);
    printf(fmt, "Message in Named memory written to Q");
/****************************************************************
    WRITE Q ITEM IN MEMORY GIVEN TO OWNER
****************************************************************/
    err = DosAllocSeg(sizeof(text2), &SharedSeg, SEG_GIVEABLE);
    if (err) errmsg("Alloc Giveable", err);
    outmsg = MAKEP(SharedSeg, 0);
    strcpy(outmsg, text2);
    err = DosGiveSeg(SharedSeg, owner, &Gseg);
    if (err) errmsg("Give Seg", err);
    outmsg = MAKEP(Gseg, 0);
    err = DosWriteQueue(Qhan, I_OWN,        /* msg in given memory */
                        sizeof(text2),      /* msg length */
                        outmsg, 11);        /* address & priority  */
    if (err) errmsg("Write Q, Givable memory", err);
    printf(fmt, "Message in Givable memory written to Q");
    DosFreeSeg(SharedSeg);
/****************************************************************
    WRITE Q ITEM IN MEMORY OWNER WILL GET
****************************************************************/
    err = DosAllocSeg(sizeof(text3), &SharedSeg, SEG_GETTABLE);
    if (err) errmsg("Alloc Gettable", err);
    outmsg = MAKEP(SharedSeg, 0);
    strcpy(outmsg, text3);
    err = DosWriteQueue(Qhan, I_DISJ,       /* msg in gettable memory */
                        sizeof(text3),      /* msg length */
                        outmsg, 11);        /* address, priority   */
    if (err) errmsg("Write Q, Gettable memory", err);
    printf(fmt, "Message in Gettable memory written to Q");
/****************************************************************
    WRITE A NUMERIC VALUE
****************************************************************/
    err = DosWriteQueue(Qhan, I_NUM, /* msg is a number */
                        1234,        /* can be anything */
                        87654321L,   /* value in lieue of address */
                        11);         /* priority */
    if (err) errmsg("Write Q, Value", err);
    printf(fmt, "Value written to Q");
/****************************************************************
    CHECK THAT ITEM IN GETTABLE MEMORY HAS BEEN READ, THEN EXIT
****************************************************************/
    items = 1;
    while (items >= 1 && !err) {
        DosSleep(100L);
        err = DosQueryQueue(Qhan, &items);
    }
    printf(fmt, "Good-bye from child process");
}
/****************************************************************
    ERRMSG: Print error message, terminate process
****************************************************************/
errmsg(char *func, USHORT err)
{
    printf("Process %d: Error %d in %s", progid, err, func);
    exit(1);
}
```

Listings can be downloaded using PCTECHline, 301/740-8383. Parameters: 2400/1200/300 bps, no parity, 8 data bits, 1 stop bit.

U.S. Postal Service
STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION
Required by 39 U.S.C. 3685)

1A. TITLE OF PUBLICATION	1B. PUBLICATION NO.	2. DATE OF FILING
PC Tech Journal	0 7 3 8 0 1 9 4	10/1/88

3. FREQUENCY OF ISSUE	3A. NO. OF ISSUES PUBLISHED ANNUALLY	3B. ANNUAL SUBSCRIPTION PRICE
Monthly	12	$34.97

4. COMPLETE MAILING ADDRESS OF KNOWN OFFICE OF PUBLICATION *(Street, City, County, State and ZIP+4 Code) (Not printers)*
One Park Avenue, New York, NY 10016

5. COMPLETE MAILING ADDRESS OF THE HEADQUARTERS OF GENERAL BUSINESS OFFICES OF THE PUBLISHER *(Not printer)*
One Park Avenue, New York, NY 10016

6. FULL NAMES AND COMPLETE MAILING ADDRESS OF PUBLISHER, EDITOR, AND MANAGING EDITOR *(This item MUST NOT be blank)*
PUBLISHER *(Name and Complete Mailing Address)*
Donald J. Byrnes 10480 Little Patuxent Pkwy Columbia, MD 21044
EDITOR *(Name and Complete Mailing Address)*
Julie Anderson 10480 Little Patuxent Pkwy, Columbia, MD 21044
MANAGING EDITOR *(Name and Complete Mailing Address)*
None

7. OWNER *(If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address, as well as that of each individual must be given. If the publication is published by a nonprofit organization, its name and address must be stated.) (Item must be completed.)*

FULL NAME	COMPLETE MAILING ADDRESS
Ziff Communications Company	One Park Avenue, New York, NY 10016
Ziff Investment Partnership, L.P.	One Park Avenue, New York, NY 10016
Philip B. Korsant	One Park Avenue, New York, NY 10016

8. KNOWN BONDHOLDERS, MORTGAGEES, AND OTHER SECURITY HOLDERS OWNING OR HOLDING 1 PERCENT OR MORE OF TOTAL AMOUNT OF BONDS, MORTGAGES OR OTHER SECURITIES *(If there are none, so state)*

FULL NAME	COMPLETE MAILING ADDRESS
None	

9. FOR COMPLETION BY NONPROFIT ORGANIZATIONS AUTHORIZED TO MAIL AT SPECIAL RATES *(Section 423.12 DMM only)*
The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes *(Check one)*

(1) ☐ HAS NOT CHANGED DURING PRECEDING 12 MONTHS
(2) ☐ HAS CHANGED DURING PRECEDING 12 MONTHS
(If changed, publisher must submit explanation of change with this statement.)

10. EXTENT AND NATURE OF CIRCULATION *(See instructions on reverse side)*	AVERAGE NO. COPIES EACH ISSUE DURING PRECEDING 12 MONTHS	ACTUAL NO. COPIES OF SINGLE ISSUE PUBLISHED NEAREST TO FILING DATE
A. TOTAL NO. COPIES *(Net Press Run)*	166,927	178,300
B. PAID AND/OR REQUESTED CIRCULATION 1 Sales through dealers and carriers, street vendors and counter sales	22,054	24,300
2. Mail Subscription *(Paid and/or requested)*	107,653	108,200
C. TOTAL PAID AND/OR REQUESTED CIRCULATION *(Sum of 10B1 and 10B2)*	129,707	132,500
D. FREE DISTRIBUTION BY MAIL, CARRIER OR OTHER MEANS SAMPLES, COMPLIMENTARY, AND OTHER FREE COPIES	5,666	7,000
E. TOTAL DISTRIBUTION *(Sum of C and D)*	135,373	139,500
F. COPIES NOT DISTRIBUTED 1. Office use, left over, unaccounted, spoiled after printing	7,056	11,100
2. Return from News Agents	24,498	27,700
G. TOTAL *(Sum of E, F1 and 2—should equal net press run shown in A)*	166,927	178,300

11. I certify that the statements made by me above are correct and complete
SIGNATURE AND TITLE OF EDITOR, PUBLISHER, BUSINESS MANAGER, OR OWNER
William L. Phillips, Vice President

Cocaine lies.

After nearly a decade of being America's glamour drug, researchers are starting to uncover the truth about cocaine.

It's emerging as a very dangerous substance.

No one thinks the things described here will ever happen to them. But you can never be certain. Whenever and however you use cocaine, you're playing Russian roulette.

You can't get addicted to cocaine.

Cocaine was once thought to be non-addictive, because users don't have the severe *physical* withdrawal symptoms of heroin—delirium, muscle-cramps, and convulsions.

However, cocaine is intensely addicting *psychologically.*

In animal studies, monkeys with unlimited access to cocaine self-administer until they die. One monkey pressed a bar 12,800 times to obtain a single dose of cocaine. Rhesus monkeys won't smoke tobacco or marijuana, but 100% will smoke cocaine, preferring it to sex and to food—even when starving.

Like monkey, like man.

If you take cocaine, you run a 10% chance of addiction. The risk is higher the younger you are, and may be as high as 50% for those who smoke cocaine. (Some crack users say they felt addicted from the *first time* they smoked.)

When you're addicted, all you think about is getting and using cocaine. Family, friends, job, home, possessions, and health become unimportant.

Because cocaine is expensive, you end up doing what all addicts do. You steal, cheat, lie, deal, sell anything and everything, including yourself. All the while you risk imprisonment. Because, never forget, cocaine is illegal.

There's no way to tell who'll become addicted. But one thing is certain.

No one who is an addict, set out to become one.

C'mon, just once can't hurt you.

Cocaine hits your heart before it hits your head. Your pulse rate rockets and your blood pressure soars. Even if you're only 15, you become a prime candidate for a heart attack, a stroke, or an epileptic-type fit.

In the brain, cocaine mainly affects a primitive part where the emotions are seated. Unfortunately, this part of the brain also controls your heart and lungs.

A big hit or a cumulative overdose may interrupt the electrical signal to your heart and lungs. They simply stop. That's how basketball player Len Bias died.

If you're unlucky the first time you do coke, your body will lack a chemical that breaks down the drug. In which case, you'll be a first time O.D. Two lines will kill you.

Sex with coke is amazing.

Cocaine's powers as a sexual stimulant have never been proved or disproved. However, the evidence seems to suggest that the drug's reputation alone serves to heighten sexual feelings. (The same thing happens in Africa, where natives swear by powdered rhinoceros horn as an aphrodisiac.)

What is certain is that continued use of cocaine leads to impotence and finally complete loss of interest in sex.

It'll make you feel great.

Cocaine makes you feel like a new man, the joke goes. The only trouble is, the first thing the new man wants is more cocaine.

It's true. After the high wears off, you may feel a little anxious, irritable, or depressed. You've got the coke blues. But fortunately, they're easy to fix, with a few more lines or another hit on the pipe.

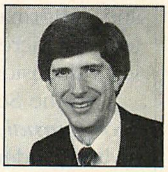
Of course, sooner or later you have to stop. Then—for days at a time—you may feel lethargic, depressed, even suicidal.

Says Dr. Arnold Washton, one of the country's leading cocaine experts: "It's impossible for the nonuser to imagine the deep, vicious depression that a cocaine addict suffers from."

Partnership for a Drug-Free America

OUTFITTING THE END USER

Burnout City Limits



P.C. Coffee

Today's user has to do too much work to produce an often misleading answer. Applications must become smarter.

Someone recently observed that the essence of modern management is to keep the entire system under a constant state of stress. Replace or redesign the parts (and the people) that break and scrutinize those that never break, because they may be a waste of resources. Does this sound like the way your office runs?

The problem is that today we can work faster on a single desktop than we can through traditional channels; further, this minimum achievable time has rapidly become the expected level of performance (what we might call the Federal Express syndrome). To achieve this level of performance on a regular basis, the competent specialist must become a one-person, cradle-to-grave project-management team.

This means that one person may be expected to retrieve data from the mainframe, compare the numbers against the business plan, identify and analyze the important differences, and produce a report and presentation for an audience two or three levels up in the organization. This person has to use applications and tools originally designed for experts who know a whole lot more about the domain that the tool supports.

Statistics packages, for example, are mainly designed for use by statisticians; when a novice fits a second-order curve to three data points and gets a perfect fit, few tools will warn the user that this is a trivial result.

Users are becoming permanent residents of Burnout City; the next generation of applications must be their survival kits. It is no longer enough for the development community to shake its collective head and mutter about the inevitable misuse of a potentially powerful tool; the time has come to use the power of today's advanced PCs to make applications smarter, competing on more than just the traditional criteria of speed and features.

ACTION IS MORE FUN

Why hasn't this happened before? Because it's more fun to build systems that *do* than systems that *know*. Putting task knowledge into a system is one of the hardest parts of computer systems development.

For example, a company called Alacritous produces a system for automated preparation of business plans. Its product, Alacrity, generates a 25-page draft business plan, which is assembled from a library of 2,900 possible recommendations. This plan is based on the results produced by a 3,000-rule expert system. Alacrity can similarly generate a draft financial forecast of 11 spreadsheets, as well as a 10-page draft report interpreting those figures.

Think about the effort required to produce a library of 2,900 recommendations so that subsets can fit together in various ways to produce a reasonably coherent piece of prose. Think about the effort required to produce those 3,000 rules that conduct a reasonably intelligent (though narrowly focused) conversation with a user who is emphatically *not* comfortable with computers.

The effort required to implement a decent forward-chaining, expert-system algorithm pales by comparison, and even the task of wrapping a high-performance user interface around the result suddenly looks trivial. (Alacritous Chairman Alistair Davidson, speaking at the Ashton-Tate Developer's Conference last September in Los Angeles, described his company's use of Ashton-Tate's Framework environment as a user-interface construction kit.)

Perhaps Alacrity is a high-tech version of the collections of prewritten business letters advertised in the back pages of management magazines; then again, are those collections really such a bad idea? In an age when the spelling checker and the thesaurus have become two of the expected features of any decent word processor, it is surprising that people are still composing routine documents from scratch. Wonderful things are in store for the first developer who succeeds in automating the construction of new documents by cutting, pasting, and smoothing together the readily applicable pieces of the user's earlier work.

Indeed, I suspect that this cut-and-paste function—performed manually—is what's really driving the power user's perceived need for ever-faster machines. Where spreadsheet recalculation time was once the principal parameter of competition, the capacity

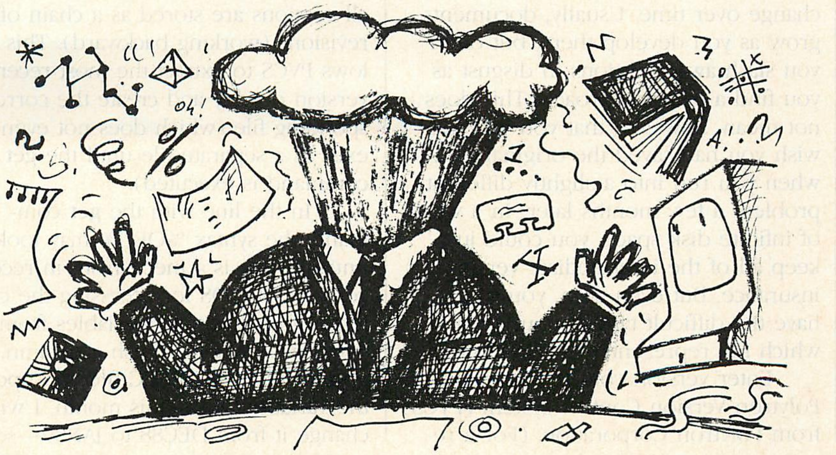


ILLUSTRATION • MACIEK ALBRECHT

and performance of the storage subsystem are increasingly becoming the critical dimension of performance.

In the September 1988 issue of *PC Tech Journal*, the reader opinion card invited readers to respond to the question of whether or not their applications really need a 25-MHz 80386-based platform. (For the results, see Professional Viewpoint, Jordene Zeimetz, this issue, p. 160.) My own response is that I primarily need access to larger, faster storage just to keep my archive search times from growing without bound. My private collection of on-line information becomes steadily larger, and with increasing use of E-mail and other services, its content becomes ever more essential to my work.

GEE, I USED TO KNOW THAT

I have a special interest in tools that make personal archives more accessible; indeed, my August 1988 column ("Long Dazed Journey into Bytes," p. 141) was devoted to the subject. Since then, I've run across two encouraging signs that indicate the problem is being addressed.

For the programmer, I see a welcome proliferation of programming environments with integral text-management tools. In Gold Hill Computing's Golden Common LISP Developer, the user can simply ask to edit a function's definition; the system automatically opens the source file from which the function was last defined, with the cursor prepositioned at that point. If you work with text, Transoft's Co-Counsel provides high-level support for finding and summarizing material on specific topics from a large collection of files; it can produce its summaries as ASCII-, WordPerfect-, or XYWrite-formatted documents.

Another problem associated with developing documents—either straight text or source code—is that they change over time. Usually, documents grow as you develop them, but often you slash away portions in disgust as you find a better approach. This does not mean, however, that you will not wish you had saved the original idea when you run into a slightly different problem a few months later. In a world of infinite disk space, you could just keep all of the intermediate versions as insurance. But even then, you would have the difficult task of figuring out which file represents which variation.

Enter version control—in my case, Polytron Version Control System (PVCS) from Polytron Corporation. (For a review of PVCS, see Product Watch, October 1988, p. 131.) I prepared this column using Borland's Sprint, which I favor because of its crash-proofing features and its user-configurable interface. I can use Sprint for everything from editing batch files to developing fully formatted documents. Sprint also has the advantage of producing straight ASCII files, lending it to use with systems such as PVCS (which are really designed to meet the needs of programmers and their collections of source code).

FIGURE 1: ***OEU Batch File***

```
echo off
cls
c:
CD \PCTJ\COLUMNS
if exist \sp\sp.swp  ren \sp\sp.swp  sp.old
if exist \sp\oeu.swp ren \sp\oeu.swp sp.swp
get \pctj\columns\%OEU%.OEU
set oldprompt=%PROMPT%
PROMPT OEU»%PROMPT%
c:\sprint\sp %1
set PROMPT=%oldprompt%
set oldprompt=
put \pctj\columns\%OEU%.OEU
if exist \sp\sp.swp  ren \sp\sp.swp oeu.swp
if exist \sp\sp.old  ren \sp\sp.old sp.swp
cd \
cls
```

Working with Sprint and PVCS, this batch file retains a chain of file revisions, always returning the cursor to the point where the user left off.

When I type OEU (for Outfitting End User) at the DOS prompt, I'm running the batch file that appears as figure 1. The first four lines are straightforward. The next two lines that begin "if exist" let me preserve the swap file that Sprint left on the disk after the last routine editing session, arranging for me to come up at the same point in the current column where I last left off.

The **get** command tells PVCS to look at the column's log file, in which all versions are stored as a chain of revisions (working backward). This allows PVCS to extract the most recent version quickly and create the corresponding file (which does not even exist as a separate file until the **get** command is executed).

In the line with the **get** command, the syntax %OEU% may look unusual; this is a mechanism in recent versions of DOS for accessing the contents of environment variables from batch files. When I finish a column, I go into the AUTOEXEC file and update the variable OEU—this month, I will change it from DEC88 to JAN89—so that the OEU batch file is aware that I am now working on the January 1989 column. PVCS and Sprint then create or reopen the proper file accordingly.

The same syntax is used to push a prefix onto my prompt string to warn me if I go into a child session of DOS. I like this technique because it works to arbitrary depth. I can export a text file from Ashton-Tate's Framework, go into a child session to edit the file with a Sprint-specific capability, then open another layer of DOS from Sprint to use my VT-100 emulator (which works better than Framework for uploads to MCI Mail). By this time, the prompt looks like

VT»Sp»FW»D:\MAILINGS>

At least I receive plenty of warning that memory may be getting tight.

Buried in the middle of the batch file is the one line that actually runs Sprint; when I quit the word processor, the remainder of the batch file goes through all of these steps in reverse. The **put** command tells PVCS to reintegrate the work file with the log file, automatically assigning the next incremental version number, deleting the work file itself, and prompting me for a description of the changes I have made. This description can be retrieved by another PVCS command, although I usually use the menu-driven PVCS shell for these tasks.

This is one of the easiest and most effective ways to add value in the DOS environment: using the capabilities of the system to integrate the functionality of discrete applications and tools into higher-level, task-oriented commands. These help the user (or developer) get a grip on the complexity of increasingly dynamic uses of information.

Of course, I would be happier if I could get such services from a collaborative network—one that could help track down related work being done by others, in addition to my own work. This is the essence of the ideal described by Mitch Kapor (formerly of Lotus Development, now heading up a new effort at ON Technology) in his banquet speech at OOPSLA '88 (the conference on object-oriented programming systems, languages, and applications) last September in San Diego. I look forward to the results of ON's research.

Lacking organizational investment in network infrastructure, however, the user has little choice but to pursue this goal one desktop at a time. The result is the aggregate purchase of far more

THE PROGRAMMER'S SHOP

helps save time, money, and cut frustrations. Compare, evaluate, and find products.

FREE Magazine!

And newsletter by respected industry opinion leader Bruce Lynch! Keep up with industry trends and events. Learn about intriguing new products and technologies. Get the informed opinion of development experts on programming strategy. 1 year (6 issues) of Programmer's Update PLUS periodic issues of The Programmer's Letter for only $50! Call for your FREE sample issue of each today!

New Discoveries

D the data language

by Caltex Software, Inc. 4GL relational data management system. 3-step data manipulation (isolate, arrange, report). Extensive file import/export. Unlimited number of data files per database. Example database online. $395

Clarion Personal Developer

by Clarion Software Corporation. Stand-alone program generator. Point-and-shoot menus, scrolling data tables, pop-up data entry forms, "look-ups", computed fields, memory variables, "hot key" procedures. Read, update data from multiple files. $139

PCX Programmer's Toolkit

by GENUS Microprogramming. Save, display, manipulate PCX format images. 35+ Assembler routines display/save buffer, file, images, set palettes, access image headers. Many compilers, integrated environments supported. Include images with database application. No royalties. With source, MASM 5.0 required. $179

386 Development Tools

	List	Ours
386 Assembler/Linker	495	399
FoxBASE+/386	595	389
High C - 386	895	Call
Hoops/386-32	575	489
Lahey Fortran F77L-EM/32	895	Call
NDP C-386	595	529
NDP Fortran-386	595	529
Paradox/386	895	639

AI-Expert Systems

CxPERT V3.0 - shell for C	395	389
Exsys Professional	795	695
Personal Consultant Plus	2950	2589

AI-Languages

A.P.T. - Active Prolog Tutor	65	49
ARITY Combination Package	1095	979
PC Scheme LISP - by TI	95	79
TransLISP PLUS - call MSC	195	119
TURBO PROLOG V2.0	150	109
Turbo Prolog Toolbox	100	69

Assemblers

MS Macro Asm	150	105
OPTASM - 75K lines/minute	125	109
Turbo Assembler/Debugger	150	105
Visible Computer 80286	100	89

BASIC & Addons

BAS_C com. - BASIC to C	375	329
Exim Toolkit	100	85
LaserPak Professional	149	119
MS QuickBASIC V4.0	99	69
QuickPak Professional	149	119

C Language - Compilers

AZTEC C86 - Commercial	499	Call
High C-286 - by MetaWare	595	Call
Lattice C - V3.3	450	289
Microsoft C 5.1 - w/CodeView	450	299
Microsoft QuickC	99	69
Turbo C - by Borland	150	105
Watcom C6.5 - highly optimized	295	269

C Language - Interactive

C-terp - by Gimpel, K&R	300	229
C Trainer - by Catalytix	124	109
Instant-C - source debug	495	359

CASE

Dan Bricklin Demo II	195	179
Interactive EasyFlow	150	129
Matrix Layout - flow chart	150	139
Show Partner-FX - demos	350	299

Call for CASE info

COBOL

MS COBOL V3.0 - OS/2	900	639
Realia COBOL	995	859

Communications Addons

C Asynch Manager - Blaise	175	139
Essential Comm Library	185	129
with Breakout Debugger	310	209
Greenleaf Comm Library	229	169

DataBase Mgmt

Advanced Revelation	950	779
CLARION - complete envt.	695	599
DataFlex - by Data Access	695	599
Eagle - by Migent	495	419
Magic PC - visual database	199	179
Paradox V2.0 List: $725	725	529
R:base for DOS	725	539

Dbase

	List	Ours
Clipper Summer '87	695	449
dBASE IV	795	529
DBXL Interpreter	199	149
FoxBASE+ - V2.1	395	259
QuickSilver Diamond	599	369

Dbase Addons

dBASE Tools for C	90	75
dGE - business graphics	195	179
FLIPPER Graphics Library	195	169
IDL Integrated Devt Library	149	149
Seek-It - query by example	100	89
SilverComm Library	150	139
Tom Rettig's Library	100	79

24-hour shipping on any product in this advertisement or we pay the freight. Just mention PTC88.

Dbase Tools

CLEAR for dBase	100	89
dBRIEF w/BRIEF	275	Call
dBug - source debugger	195	179
dBX - dBASE III to C	550	529
Documentor	295	239
FoxTool Box	295	279
Genifer - code generator	395	259
R&R Relational Report	149	139
R&R Clipper & FoxBASE	55	45
UI Programmer	295	249

Debuggers

Periscope II - breakout switch	175	139
Periscope III - 10 MHz version	1395	1129
Pfix-86 Plus	395	219

Development Tools

Lattic Compiler Companion	100	Call
MKS AWK	99	89
PC - Lint	139	109
PC - Metric - analyze	199	189
Plink 86 Plus - overlays	495	299
PolyMake	149	135
PVCS Corporate	395	369
Seidl Version Manager	300	279

Editors

BRIEF	195	Call
Edix	195	159
Epsilon - like EMACS	195	159
KEDIT - like XEDIT, V.4	150	129
Personal Rexx	125	109
ME Macro Editor - Source	189	159

Editors Cont.

	List	Ours
MKS VI	149	129
PC/EDT - macros	295	269
SPF/PC - V2.0	245	189
Vedit Plus	185	139

File Addons

Btrieve ISAM	245	185
Btrieve/N - multiuser	595	455
Report Option	145	119
XQL - SQL for Btrieve	795	699
Xtrieve	245	195
c-tree by Faircom - source	395	319
d-tree - data dictionary	495	429
r-tree - report generator	295	239
c-tree - w/r-tree	650	519
CBTREE - Source	159	129
dBC ISAM III - by Lattice	250	169
dB_FILE - network model	varies	Call
dB_RETRIEVE - SQL	varies	Call
Opt Tech Sort	149	99
WKS Library	195	179

FORTRAN

FOR_C - FORTRAN '77 to C	750	679
I/O Pro - screens	149	129
Lahey FORTRAN F77L	477	459
Lahey Personal FORTRAN	95	89
MS FORTRAN - CodeView	450	299
RM/FORTRAN	750	479
Scientific Subroutines	175	125
Spindrift - DOS, windows	149	129

General Addons

C Tools Plus - V5.0	129	99
C Utilities - by Essential	185	129
Greenleaf SuperFunctions	265	179

Graphic Addons

Drawbridge	129	109
Essential Graphics	299	229
with source	598	519
GraphiC - like DISSPLA	395	309
GSS Dev't Toolkit	495	409
Halo '88 - 140+ devices	325	229
Hoops 3D Graphics	575	549
MetaWINDOW/PLUS	275	219
PCX Publishing's Toolkit	90	85
with source	190	179
QuickWindows/C	95	79

Macintosh

	List	Ours
FoxBASE+ Dev't Package	395	339
McMax by Nantucket	295	269
Lightspeed C	175	159

Call for more Mac info

Obj.-Oriented

Actor	495	449
C-Talk	150	139
Smalltalk/V	100	85
Smalltalk/V 286	200	179
Zortech C++ - true compiler	100	99

OS - Support

DESQview	130	115
MS Windows/286	99	69
MS Windows/386	195	139
Development Toolkit	500	329
MKS Toolkit - Unix shell	199	179
OS/2 Programmer's T'kit	350	249
PC/MOS 386 5 User	595	529
Timeslicer	295	265
VM/386 Multitasker	245	229
XTree Pro	129	109

Other Languages

APL Plus - by STSC	695	549
CCSM Mumps - single	160	149
Modula-2 Dev. System	249	209
RPG II complete	1400	1299
SNOBOL4+ - strings	95	85

Other Products

Back-It	130	119
CO/SESSION	249	209
Crosstalk XVI	195	129
Disk Technician Plus	130	119
Link + Locate++ - ROM MSC	395	349
Mace Utilities	99	85
MathCad	349	289
Norton Utilities Advanced	150	109
PC Tools Deluxe	80	75
SoftProbe II/TX - debug	395	345
SPSS/PC Plus	795	749
TK Solver Plus	395	339
Watchdog - security	295	265
WordPerfect	495	299
Zap - file xfer	95	89

Text Screen

C Worthy w/forms	295	Call
with source	495	Call
Curses - by Aspen Scientific	119	109
Greenleaf DataWindows	295	219
PowerScreens - by Blaise	129	99
Vitamin C - source, menus	225	169
VC Screen - painter	149	125
Windows for Data	295	265
with source, OS/2 or Xenix	590	Call

Unix/Xenix

DosMerge 286	249	229
DosMerge 386	499	479
Edix - editor	275	249
JAM JYACC Applic. Mgr.	many	Call
Microport System V/286	898	795
Microport System V/386	1198	999
SCO Xenix System V 286	1295	999
SCO Xenix System V 386	1595	1439
Windows for Data	795	719

Note: Mention this ad. Some prices are specials. Ask about COD and POs. Formats 3" laptop now available, plus 200 others. UPS surface shipping add $3/per normal item. All prices subject to change without notice.

Established 1983. We carry over 1,100 *different* products. Call today for complete technical information and advice, our catalog, literature, and *solid value.*

5-P Pond Park Road
Hingham, MA 02043
Mass: 800-442-8070
or 617-740-2510 10/88
Telex: 671-5348
FAX: 749-2018

800-421-8006

HOURS:
M-F 8:30 - 8:00
Sat: 10:00 - 4:00

processing power and storage capacity than a collaborative system would necessarily use—this is unfortunate, expensive, and far less effective than task-oriented and group-oriented systems should be.

Networks are an attractive subject, but they cannot be controlled from the development side of the fence. There are other ways, however, in which systems can work harder to know and to learn, thereby giving more genuine help to the user and making life a little easier in Burnout City.

LOCAL OPTIONS

If you want to identify opportunities to make your applications smarter, look at the books (*other* than your own documentation) to which your user has to refer while using your application. I call these task-related references the "wetware documentation," from the slang term for the human brain used by researchers in the artificial intelligence community.

For example, writers often keep a copy of Strunk and White's *The Elements of Style* close at hand. This is a useful tool for resolving annoying questions, such as: "Should this be a colon here, or a semicolon?" and "What is the correct word to use here: *which* or *that*?"

A user would benefit if a hot-key combination, such as Alt-?, could do an associative retrieval of the relevant portion of Strunk and White's book, based on the cursor's current position: for example, usage of the colon and semicolon if you are at either of those marks, or comments on tricky words (*which* versus *that*, *can* versus *may*) at which you are currently located.

If you think I'm off in fantasy land by suggesting a product that demands CD-ROM or some other high-tech delivery system to be practical, perhaps I should mention that the entire text of my copy of *The Elements of Style* is only 85 pages in length. For the programmer, the text of Kernighan and Ritchie's standard reference, *The C Programming Language*, is only 219 pages in length, or about 560KB. Storage space is not the issue.

Another handy service for the end user would be a system that warns of probable errors based on simple statistical criteria. A poor little algorithm may have difficulty deciding when you have improperly used *which* instead of *that*, but it should have no problem analyzing a reasonably large sample of text and getting an idea of the normal proportions in which those words usually occur. A program might then be able to say, "You are using *which* twice as often as *that* when the normal ratio is [whatever]; would you like to review the usage of this word?"

ERRORS AND EXPECTATIONS

Enough about text—even though words represent almost a third of the output of even the most technical professional. Let's take a look at spreadsheets and data managers as well.

In the real world, almost everyone derives competitive advantage from making more accurate predictions about what will happen under previously unknown conditions: a greater load on a bridge, a change in the temperature of a chemical reaction, or the passage of time. Such predictions are often extrapolations of data, and many modern spreadsheets provide tools for regression calculations to assist in the line-fitting process.

Almost unknown, however, are the procedures for putting error bands on regressions. It is not theoretically difficult, merely tiresome, but the resulting hyperbolic curves—which diverge rapidly as one goes beyond the range of past experience—demonstrate the high risk of extrapolation in a way that commands attention.

Tools are beginning to appear that allow spreadsheet-like manipulation of uncertain quantities; I've tried to implement these features, and they use up CPU resources at an unbelievable rate, so I don't expect them to become mainstream facilities in the near future. The automatic provision of uncertainty bands on regressions, however, can easily be done on any platform capable of running a full-blown spreadsheet program. The development community has an ethical responsibility to help the user assign an appropriate level of caution to the results our programs so easily generate. Many more opportunities in the quantitative applications arena are possible.

Data-management tools also could do much more to help the end user. For example, about two weeks ago, I was assembling a mailing list for a membership drive on behalf of the Personal Computer Professionals Association. As I was transcribing names and addresses from a list of the 100 largest public companies in Los Angeles County, the 25 largest high-technology firms, and so on, I realized that my file manager could be doing more to help me out. When I was in the City field of the database and typed *Bev*, the rest of the entry in every case was *erly Hills*; similarly, a street name beginning with *Cen* was almost always concluded with *tury Boulevard*. Why did I have to decide for myself when an entry was occurring often enough to merit a macro, let alone have to manage for myself the name conflicts that always occur when your vocabulary is limited to Alt-A through Alt-Z?

The one feature I would like most to add to any program with reasonably structured input is this kind of limited pattern recognition—with a user interface that can offer to complete the entry without getting in the way. Perhaps this is finally a reasonable use for the SysReq key that's been wasting space on the keyboard for all these years. Let the system give some sign that it knows what's coming—an indicator on the status line, for example—and let the SysReq key trigger a display of what the program thinks the user might want next.

Using the Esc key could mean "don't bother," while hitting the SysReq key a second time would say, "fine, do it." Instead of typing *Beverly Hills*, a more likely sequence would then be *Bev*{SysReq}{SysReq}. Time savings would add up quickly.

MARGIN RELEASE

If one common theme is present in corporate America today, it's the ever-decreasing margins for error. Uncertainties are estimated on an itemized basis, with overall factors of safety greatly reduced in the pursuit of competitive designs. Just-in-time inventory management in the factory makes people go to great lengths to understand what is happening and when—at a level of detail that has rarely been attempted in the past.

To meet these requirements, our end users have to be able to respond more rapidly and demonstrate more versatility than ever before. If we want our applications to be an essential part of tomorrow's work environment, we must build personal support systems that integrate a correspondingly broader set of skills: knowledge, not just features; power, not just speed. These are good goals for your development efforts in 1989.

Peter C. Coffee is managing partner of SolveWare, a developer and business computing consultant, and is active in AI and distributed computing applications for aerospace and educational clients.

C CODE FOR THE PC

source code, of course

	MS-DOS File Compatibility Package (create, read, & write MS-DOS file systems on non-MS-DOS computers)	$500
	Bluestreak Plus Communications (two ports, programmer's interface, terminal emulation)	$400
	CQL Query System (SQL retrievals plus windows)	$325
	GraphiC 4.1 (high-resolution, DISSPLA-style scientific plots in color & hardcopy)	$325
	PC Curses (Aspen, Software, System V compatible, extensive documentation)	$250
NEW!	Greenleaf Business Mathlib (exact decimal math, formatting, depreciation, interest, cash flow, statistics)	$250
	Greenleaf Data Windows (windows, menus, data entry, interactive form design)	$220
	Greenleaf Communications Library (interrupt mode, modem control, XON-XOFF)	$175
	TurboTEX (TRIP certified; HP, PS, dot drivers; CM fonts; LaTEX)	$170
NEW!	Sherlock (C debugging aid)	$170
	Essential resident C (TSRify C programs, DOS shared libraries)	$165
	Greenleaf Functions (296 useful C functions, all DOS services)	$160
	Essential C Utility Library (400 useful C functions)	$160
	Essential Communications Library (C functions for RS-232-based communication systems)	$160
	WKS Library Version 2.0 (C program interface to Lotus 1-2-3, dBase, Supercalc 4, Quatro, & Clipper)	$155
	OS/88 (U**x-like operating system, many tools, cross-development from MS-DOS)	$150
	ME Version 2.0 (programmer's editor with C-like macro language by Magma Software; Version 1.31 still $75)	$140
	Turbo G Graphics Library (all popular adapters, hidden line removal)	$135
	Vmem/C (virtual memory manager; least-recently used pager; dynamic expansion of swap file)	$130
	TurboGeometry (library of routines for computational geometry)	$125
	CBTree (B+tree ISAM driver, multiple variable-length keys)	$115
	Minix Operating System (U**x-like operating system, includes manual)	$105
	PC/IP (CMU/MIT TCP/IP implementation for PCs)	$100
	B-Tree Library & ISAM Driver (file system utilities by Softfocus)	$100
NEW!	Tele Operating System (TeleKernel, TeleWindows, TeleFile, & TeleBTree by Ken Berry)	$100
	The Profiler (program execution profile tool)	$100
	QC88 C compiler (ASM output, small model, no longs, floats or bit fields, 80+ function library)	$90
	Wendin Operating System Construction Kit or PCNX, PCVMS O/S Shells	$80
	C Windows Toolkit (pop-up, pull-down, spreadsheet, CGA/EGA/Hercules)	$80
	JATE Async Terminal Emulator (includes file transfer and menu subsystem)	$80
NEW!	Polyglot Lisp-to-C Translator (includes Lisp interpreter, Prolog, and simple calculus prover)	$80
	MultiDOS Plus (DOS-based multitasking, intertask messaging, semaphores)	$80
NEW!	Kinetic Image Synthesizer (3-D animation system ... Saturday morning on your PC!)	$75
	XT BIOS Kit (roll your own BIOS with this complete set of basic input/output functions for XTs)	$75
NEW!	TE Editor Developer's Kit (full screen editor, undo command, multiple windows)	$75
	Professional C Windows (lean & mean window and keyboard handler)	$70
	lp (flexible printer driver; most popular printers supported)	$65
	Quincy (interactive C interpreter)	$60
	PTree (parse tree management)	$60
NEW!	Icon-Tools (full-featured icon display and editing system)	$60
NEW!	Polyglot TSR Package (includes reminder, bookmark, virus catcher, cache manager, & speech generator)	$50
	HELP! (pop-up help system builder)	$50
	Multi-User BBS (chat, mail, menus, sysop displays; uses Galacticomm modem card)	$50
	Make (macros, all languages, built-in rules)	$50
	Coder's Prolog (inference engine for use with C programs)	$45
	Virtual Memory System (least recently used swapping)	$40
	C-Notes (pop-up help for C programmers ... add your own notes)	$40
NEW!	Heap I/O (treat all or part of a disk file as heap storage)	$40
	Biggerstaff's System Tools (multi-tasking window manager kit)	$40
	PC-XINU (Comer's XINU operating system for PC)	$35
	CLIPS (rule-based expert system generator, Version 4.2)	$35
	Tiny Curses (Berkeley curses package)	$35
NEW!	Polyglot RAM Disk (change disk size on the fly; includes utilities)	$30
	SP (spelling checker with dictionary and maintenance tools)	$30
	Clisp (Lisp interpreter with extensive internals documentation)	$30
	Translate Rules to C (YACC-like function generator for rule-based systems)	$30
	6-Pack of Editors (six public domain editors for use, study & hacking)	$30
	Crunch Pack (14 file compression & expansion programs)	$30
NEW!	Pascal Compiler & Interpreter (P-codes, standard Pascal)	$25
	ICON (string and list processing language, Version 7)	$25
	FLEX (fast lexical analyzer generator; new, improved LEX)	$25
	LEX (lexical analyzer generator; an oldie but a goodie)	$25
	Bison & PREP (YACC workalike parser generator & attribute grammar preprocessor)	$25
	AutoTrace (program tracer and memory trasher catcher)	$25
NEW!	Data Handling Utilities in C (data entry, validation & display; specify Turbo C or Microsoft)	$25
	Arrays for C (macro package to ease handling of arrays)	$25
NEW!	ANSI Forms (forms manager based on ANSI codes)	$20
	C Compiler Torture Test (checks a C compiler against K & R)	$20
	Benchmark Package (C compiler, PC hardware, and Unix system)	$20
	A68 (68000 cross-assembler)	$20
	List-Pac (C functions for lists, stacks, and queues)	$20
	XLT Macro Processor (general purpose text translator)	$20
NEW!	KeySwap (swaps CapsLock and CRTL key functions on XTs and ATs)	$20
NEW!	OOPS (collection of handy C++ classes by Keith Gorlen of NIH)	$20
	C/reativity (Eliza-based notetaker)	$15

Data

DNA Sequences (GenBank 55.0 including fast similarity search program)	$150
Protein Sequences (5,415 sequences, 1,302,966 residuals, with similarity search program)	$60
Dictionary Words (234,932 words in alphabetical order, no definitions)	$60
U. S. Cities (names & longitude/latitude of 32,000 U.S. cities and 6,000 state boundary points)	$35
The World Digitized (100,000 longitude/latitude of world country boundaries)	$30
KST Fonts (13,200 characters in 139 mixed fonts: specify TEX or bitmap format)	$30
USNO Floppy Almanac (high-precision moon, sun, planet & star positions)	$20
NBS Hershey Fonts (1,377 stroke characters in 14 fonts)	$15
U. S. Map (15,701 points of state boundaries)	$15

The Austin Code Works
11100 Leafwood Lane
Austin, Texas 78750-3409 USA

acw!info@uunet.uu.net

Voice: (512) 258-0785
BBS: (512) 258-8831
FAX: (512) 258-1342

Free surface shipping on prepaid orders **For delivery in Texas add 7%** **MasterCard/VISA**

PARADISE PRICES 1-800-445-7899

CALL PROGRAMMER'S PARADISE TODAY

and discover our commitment to bringing you the best software at the best prices. You'll also find software pros to help you select the product that you need plus immediate shipment on our stock of 1000 products and a 30-day money back guarantee. Paradise is ready and waiting for you.

386 SOFTWARE

	LIST	OURS
386/ASM/LINK	495	399
386MAX	75	66
386MAX PROFESSIONAL	129	115
DESQview	130	115
High C 386	895	799
Microport Sys. V/386 (Complete)	899	799
Microsoft Windows/386	195	130
NDP-C or Fortran-386	595	529
NDP FORTRAN-386	595	529
SCO 386 XENIX Sys. V (Complete)	1495	1195
VM/386	245	199

ASSEMBLY LANGUAGE

	LIST	OURS
Advantage Disassembler	295	279
MS Macro Assembler	150	99
Turbo Assembler/Debugger	150	105

NEW RELEASES

386MAX PROFESSIONAL
Combines the popular 386MAX and new 386LOAD for total 386 memory management. Ideal tool for relocating network and other devise drivers.
List: $129 **Ours: $115**

SCRIMAGE
Scrimage's new screen and menu design tool works in the format of the language you are using, without the overhead of a memory resident utility. Use Scrimage with dBASE III Plus, IV, dBASE Compilers, C or BASIC.
List: $99 **Ours: $90**

MKS LEX & YACC
New 100% UNIX compatible compiler construction tools for the DOS C programmer. MKS LEX & YACC work together efficiently to simplify the creation of new languages for applications, or to develop new programming language compilers.
List: $249 **Ours: $212**

BASIC

	LIST	OURS
MS BASIC/6.0	295	199
MS QuickBASIC	99	69
QuickPak Professional	149	129
QuickWindows (w/source)	99	90
Turbo Basic	100	69

C LANGUAGE

	LIST	OURS
ADVANTAGE C++	495	479
C-terp	298	232
Lattice C	450	289
Microsoft C	450	299
QuickC	99	69
w/serial mouse	249	149
RUN/C Professional	250	159
Turbo C v. 2.0	150	105
Turbo C 2.0 Professional	250	175

C LIBRARIES/UTILITIES

	LIST	OURS
C ASYNCH MANAGER	175	137
C TOOLS PLUS/5.0	129	99
C Utility Library	199	139
Essential Communications	185	121
Greenleaf Bus. MathLib	325	229
Greenleaf Comm Library	229	169
Greenleaf Functions	209	155
Greenleaf SuperFunctions	265	189
Greenleaf TurboFunctions	109	79
PC-lint	139	101
PforCe	495	215
Pre-C	295	159
TimeSlicer	295	279
Turbo C TOOLS	129	99

GRAPHICS

	LIST	OURS
Essential Graphics	299	229
GraphiC	395	322
HALO '88	325	229
TurboHALO	100	80

SCREENS/WINDOWS

	LIST	OURS
Greenleaf Data Windows	295	209
MS Windows Dev. Toolkit	500	319
MS OS/2 Prog. Toolkit	350	229
PANEL Plus	495	395
PANEL/QC or /TC	129	99
Vitamin C	225	162
VCScreen	149	109
Windows for Data	295	259

FORTRAN

	LIST	OURS
F77L	477	429
Grafmatic or Plotmatic	135	119
Lahey Personal FORTRAN 77	95	89
MS FORTRAN	450	299
Spindrift Library	149	135

PASCAL COMPILERS

	LIST	OURS
Microsoft Pascal	300	189
Turbo Pascal v. 5.0	150	105
Turbo Pascal 5.0 Prof.	250	175

TURBO PASCAL ADD-ONS

	LIST	OURS
DATABOSS APPLIC. GEN.	399	359
Overlay Manager	45	39
Pascal ASYNCH MANAGER	175	137
T-Debug Plus v. 4.0	45	40
Turbo ASYNCH PLUS	129	99
Turbo Geometry Library	150	135
TurboMAGIC	199	179
Turbo POWER SCREEN	129	99
Turbo POWER TOOLS PLUS	129	99
Turbo Professional 4.0	99	80

EDITORS

	LIST	OURS
BRIEF	195	CALL
KEDIT	150	120
MKS/VI	149	127
Norton Editor	75	70
SPF/PC	245	185
VEDIT PLUS 3.0	185	115

FILE MANAGEMENT

	LIST	OURS
Btrieve	245	185
Xtrieve	245	189
Report Option	145	109
Btrieve/N	595	455
CBTREE	159	141
c-tree	395	318
d-tree	495	395
r-tree	295	241
dBC III PLUS	750	599
dB_RETRIEVE	395	322
dB_FILE	395	322

1000 BRAND NAME PRODUCTS IN STOCK

	LIST	OURS
Informix Products	CALL	CALL
XQL	795	599

OBJECT-ORIENTED PROGRAMMING

	LIST	OURS
ACTOR	495	423
ADVANTAGE C++	495	479
Smalltalk/V	100	85
Communications	50	45
EGA/VGA Color Ext.	50	45
Goodies #1, #2 or #3	50	45
Smalltalk/V 286	200	169

OPERATING SYSTEMS

	LIST	OURS
Concurrent DOS 386	395	349
Microport Sys. V/AT (comp)	649	549
SANTA CRUZ OPER. (SCO):		
XENIX Sys V (comp)	1295	999
Wendin-DOS	139	109

OTHER PRODUCTS

	LIST	OURS
Dan Bricklin's Prog. II	195	179
EASYFLOW	150	135
Norton On-Line Program Guides	100	75
Norton Utilities, Advanced	150	101
Opt-Tech Sort	149	105
PC/Tools Deluxe	80	70
Periscope I	695	556
Periscope III 10 MHz	1495	1196
Plink86plus	495	279
PolyAWK	99	79
PolyDoc	199	159
PVCS Corporate	395	322

HOW WE WORK

PHONE ORDERS

Hours 9 AM–7 PM EST. We accept MasterCard, Visa, American Express. Include $3.95 per item for shipping and handling. All shipments by UPS ground. Rush service available, ask for rates when you order.

MAIL ORDERS

POs by mail or fax are welcome. Please include phone number.

TECHNICAL SUPPORT FROM SOFTWARE PROS

RETURN POLICY

Our "No Hassle" policy means if you're not satisfied with a product from Programmer's, simply return it within 30 days for a refund. Some manufacturer's products cannot be returned once disk seals are broken, so check before you buy.

INTERNATIONAL SERVICE

Take advantage of our International business number for details on export charges and exchange rates. Payments should be made in U.S. dollars.

DEALERS AND CORPORATE ACCOUNTS

Call and ask for our catalog and special volume discounts.

UNBEATABLE PRICES

We'll match lower nationally advertised prices.

In NY: **914-332-4548**
Customer Service: **914-332-0869**
International Orders: **914-332-4548**
Telex: **510-601-7602**
Fax: **914-332-4021**

Call or Write for Latest Free Catalog!

1-800-445-7899

Programmer's Paradise™

A Division of Magellan Software Corp
55 South Broadway, Tarrytown, NY 10591

CIRCLE NO. 173 ON READER SERVICE CARD

VAX Power - PC Price!

If you need or are accustomed to the throughput of a 32-bit mini, including any of DEC's VAX series, MicroWay has great news for you. The combination of our NDP compilers and our mW1167 numeric coprocessor gives VAX speed to your 386 PC! If you don't own a 386 PC, we provide a number of powerful PC and AT upgrade paths.

MicroWay offers transputer based parallel processing boards and languages for the PC, AT, or 386. Each T800 RISC processor on these boards packs the power of a 20 MHz 386/1167. A Quadputer, with four T800s, boasts 40MIPS/6 megaflops of throughput.

Many NDP Fortran-386 users are reporting turnaround times that are two to six times faster than their VAX. They are a function of the VAX processor being used, the speed of the 386, the number of users served by the VAX, and the coprocessor being used with the 386.

Dr. Robert Atwell, leading defense scientist, calculates that NDP Fortran-386 is saving him $12,000 per month in rentals of VAX hardware and software while doubling his productivity!

Fred Ziegler of AspenTech in Cambridge, Mass. reports, "I ported 900,000 lines of Fortran source in two weeks without a single problem!" AspenTech's Chemical Modeling System is in use on mainframes worldwide and is probably the largest application to ever run on an Intel processor.

Dr. Jerry Ginsberg of Georgia Tech reports, "My problems run a factor of six faster using NDP Fortran-386 on an mW1167 equipped 386/20 than they do on my MicroVAX II."

Our NDP compilers and NDP utilities are the key to taking advantage of the two to five-fold increase in speed that the mW1167 provides. The compilers generate mainframe style code, while our utilities simplify ports from either the PC or the mainframe world. A new utility enables our NDP compilers to call HALO 88.

If you are curious about the benefits of the NDP/mW1167 approach, consider the following (price estimates are for complete systems):

Coprocessor	Speed (Flops)	Price
80287-10	80,000	$3,000
80387-20	440,000	$5,000
mW1167-20	**2,100,000**	**$6,000**

When driven by a 32-bit compiler, the mW1167 approaches the speed of a $600,000 VAX 8650! In fact, many of our users have reported increases in turnaround of 2 to 4 times that of their VAX! Call us for complete details.

32-Bit Compilers and Tools

NDP Fortran-386™ and **NDP C-386™** Compilers generate globally optimized, mainframe quality code. Both run in 386 protected mode under Phar Lap extended MS-DOS, UNIX, or XENIX. The memory model employed uses 2 segments, each of which can be up to 4 gigabytes. They generate code for the 80287, 80387, or mW1167. Both include high speed EGA graphics extensions written in C that perform BASIC-like screen operations.

- **NDP Fortran-386™** Full implementation of FORTRAN-77 with Berkeley 4.2, VAX/VMS and Fortran-66 extensions. $595
- **NDP C-386™** Full implementation of AT&T's PCC with MS and ANSI extensions. . . . $595

NDP Package Pricing:

387FastPAK: NDP Compiler, Phar Lap and 80387 Coprocessor	16 MHz:	$1299
	20 MHz:	$1499
1167FastPAK: NDP Compiler, Phar Lap and mW1167 Coprocessor	16 MHz:	$1695
	20 MHz:	$2295
Phar Lap Development Tools		$495
Virtual Memory Extension		$295

NDP Windows™ — NDP Windows includes 80 functions that let you create, store, and recall menus and windows. It works with NDP C-386 and drives all the popular graphics adapters: . . . . Library: $125, C Source: $250

NDP Plot™ — Calcomp compatible plot package that is callable from NDP Fortran. It includes drivers for the most popular plotters and printers and works with CGA, Hercules, EGA and VGA $325

NDP/FFT™ — Includes 40 fast running, hand coded algorithms for single and double dimensioned FFTs which take advantage of the 32-bit addressing of the 386 or your hard disk. Callable from NDP Fortran with mW1167 and 80387 support $250
387FFT for 16-bit compilers $250

HALO 88 to NDP Graphics Interface — This module enables you to call graphics routines in HALO 88 from NDP Fortran or C $100

MicroWay® 80386 Support (508) 746-7341

Parallel Processing

Monoputer 2™
The world's most popular PC transputer development product now extends the memory available for developing transputer applications from 2 to 16 megabytes. The board now features a DMA bus interface for fast I/O. Monoputer 2, an ideal platform for porting mainframe Fortran or C code, provides the speed of a 386/20 for 20% of the cost!
Monoputer 2 with T414 (0 MB) $995
Monoputer 2 with T800 (0 MB) $1495

Quadputer™
This board for the XT, AT, or 386 can be purchased with 2, 3 or 4 transputers and 1, 4 or 8 megabytes of memory per transputer. Two or more Quadputers can be linked together to build networks with mainframe power which use up to 100 or more transputers. One customer's application has gone from 8 hours on a mainframe to 16 minutes on a system containing five Quadputers . . . . from $3495

Transputer Compilers and Applications
MicroWay offers Parallel languages for the Monoputer and Quadputer.
Logical Systems Parallel C $595
MicroWay Occam2 . . . . $495
3L Parallel C $895
3L Parallel Fortran $895
MicroWay Prolog Interpreter $750
Microfield - Finite element analysis . . $1600
ParaSoft: Parallel Environment $300
Performance Monitor $200
C Source Level Debugger $300
T800/NAG™ (See NDP/NAG) $2750

387BASIC™ — Our 16-bit MS compatible compiler introduces numeric register variables to produce the fastest running 80x87 code on the market $249

Compaq 386/20/25 Add-Ons

RAMpak™ - One megabyte 32-bit memory module fits in Compaq memory slot . . CALL
mW1167™ is built at MicroWay using Weitek components and includes an 80387 socket.
mW1167-16 . . . .$995
mW1167-20 . . . .$1595
mW1167 Microchannel-16 . . . . $1295
mW1167 Microchannel-20 . . . . $1595
Weitek 3167-25 . . . . $2495
80387-20 . . . .$595
80387-25 . . . .$695

Numeric Coprocessors

8087 . . . .$99
8087-2 . . . .$145
80287-8 . . . .$239
80287-10 . . . .$279
80387-16 . . . .$425
80387-16SX $450
80387-20 . . . .$595
80387-25 . . . .$695
287Turbo-12 (for AT compatibles) $450
256K 100ns DRAM $13
256K SIMMS $119
1 MB SIMMS $450
(All of our Intel coprocessors include 87Test.)

Intelligent Serial Controllers

MicroWay's **AT4™**, **AT8™**, and **AT16™**, the fastest intelligent serial controllers, run in AT, 80386 and PS/2 PCs. They come with drivers for UNIX, XENIX, and PC MOS.
AT4 ... $795 **AT8** ... $995 **AT16** ... $1295

32-Bit Applications

PSTAT-386 — Popular mainframe statistics package. The full version was ported . . . $1495

NDP/NAG™ — Features a library of 268 engineering and scientific numerical algorithms. Callable from NDP or 3L Fortran $895

12 MHz PC Accelerators

FastCACHE-286 12 MHz $299
SuperCACHE-286 12 MHz $399
FastCACHE-286 9 MHz $199

Micro Way

World Leader in PC Numerics

P.O. Box 79, Kingston, MA 02364 USA (508) 746-7341
32 High St., Kingston-Upon-Thames, U.K., 01-541-5466
USA FAX 617-934-2414 Australia 02-439-8400 Germany 069-75-1428

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PRODUCT CATEGORIES

ADVERTISING RATES AND INFORMATION

PC Tech Journal Marketplace is a special economical section for product and service listings.

Listings are grouped by category and sold by column inches. Second color option available.

Standard Directory Listings are also available for a minimum of 3 issues at $220.00 per issue ($660 total).

For additional information call 212-503-5115 or 800-825-4ADS.

ACCOUNT MANAGERS

Michael J. King, Account Representative
(212) 503-5116

WA, HI, AZ, MT, NV, ID, ALL CA, NM, UT, CO, OR, WY, Canada

Lee Uniacke, Account Representative
(212) 503-5141

AK, AL, AR, DC, DE, FC, FL, GA, IA, IL, IN, KS, KY, LA, MD, MI, MN, MO, MS, NB, NC, ND, NJ, OH, OK, PA, SC, SD, TN, TX, VA, WI, WV, CT, MA, ME, NH, NY, RI, VT, British Columbia (All Overseas Calls)

CLASSIFIED ADVERTISING STAFF

One Park Avenue, New York, NY 10016
(212) 503-5115

Group Sales Director
Kathryn J. Cumberlander

Group Sales Manager
Daniel L. Rosensweig

Sales Manager
Phyllis Kenny

Advertising Coordinator
Elizabeth Lopez

Advertising Support Manager
Monica Dixon

Production Director
Anne R. Brockinton

Production Coordinator
Ann Sheridan

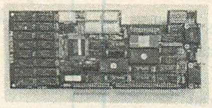
TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

HARDWARE/ACCESSORY CARDS

PEACOCK™ VGA - 16 BIT

Allstar's new Peacock™ VGA PLUS gives you SIXTEEN BIT all mode performance. Offers IBM register-to-register compatibility with BOTH TTL and Analog connectors. PS-2 Feature Connector with 512 K High Speed RAM and Five crystal controlled frequencies. Device drivers are provided for GEM, Autocad V-9, Lotus 1 2 3, Ventura, Windows 386 and others.


Made in USA

PEACOCK™ VGA Plus Does It All —

- VGA +; VGA; EGA; CGA; MDA; MCGA; & Hercules
- 16 BIT Performance (Also XT Compatible)
- 300 x 200/256; 640 x 480/256; 800 x 600/16; 800 x 600/256; 1024 x 768/16 Colors
- Both Analog and TTL Monitors Supported
- 512k High Speed RAM
- 80 COL x 66 ROWS Plus 132 COL x 44 ROWS
- Interlaced and non-Interlaced Supported

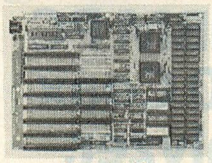
List: $595

See us at COMDEX/Fall '88

TRANSFORMER FOR PC & XT

Put the "Real Thing" in your older IBM PC or XT!

The Transformer™ 286-12 MHZ is a direct form, fit, and function, replacement board. Offered in two versions: 5 slots for PC's, 8 slots for XT's, plug compatible with all present PC/XT cards, operating devices, power supply, controllers, drives, including keyboards.


Made in USA

Converts your old IBM PC or XT into a true 12 MHZ 16 BIT AT BUS system.

Suggested Retail $495

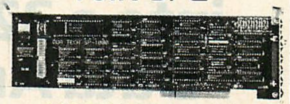
- Memory partitions: 512K • 640K • 1 MB
- OS/2, UNIX, Novell, and Token Ring Compatible
- Norton SI: 1 WAIT-13.3
- AWARD Bios with ROM setup
- ZYMOS/INTEL chip set
- 6/12 MHZ, HW/SW selectable
- 30 Day Money-Back Guarantee, 1 yr. Warranty
- $50.00 Trade-in On Old Board

ALLSTAR

Allstar Microsystems Corp.
13885 Alton Parkway, Irvine, CA 92718 • (714) 951-1884 • FAX (714) 951-2897 • **(800) 525-2860**

CIRCLE 265 ON READER SERVICE CARD

NEW PARALLEL PORT FOR PS/2



MODELS 50, 60, 80

- LPT1, LPT2, LPT3
- Optional Serial Port
- OEM Pricing Available

1-800-553-1170

QUA TECH INCORPORATED

478 E. Exchange St., Akron, OH 44304
TEL (216) 434-3154 FAX (216) 434-1409
TLX 5101012726

CIRCLE 267 ON READER SERVICE CARD

Tech Marketplace
To place your ad
Call (212) 503-5115

DYNA 386 30MHZ CACHE MOTHER BOARD WITH 64K INTELLIGENT HIGH SPEED CACHE. THE FASTEST 386 BOARD ON THE OPEN MARKET.

BENEFITS:	FEATURES:
SI RATING (ver 4.0) 36.3 SPEED TEST (ver 1.10) 49	INTEL 80386-25 CPU
DOS AND MS OS/2 COMPATIBLE	64K CUSTOM MEMORY CACHE CONTROLLER
GREAT FOR ENGINEERING WORK STATIONS	10MHZ BUS SPEED
UPGRADE YOUR AT™	UP TO 16MB OF 32 BIT MEMORY
DEAL WITH A COMPANY THAT HAS 7 YEARS IN PC BOARDS	FIVE 16 BIT SLOTS THREE 8 BIT SLOTS
WIDE RANGE OF BOARDS, SYSTEMS AND PERIPHERALS	WEITEK 3167 SUPPORT 80387 SUPPORT
KNOWLEDGEABLE STAFF	
EXCELLENT SERVICE	
LEASING AVAILABLE	

MS OS/2 IS A TRADE MARK OF MICROSOFT CORP. AT IS A REGISTERED TRADE MARK OF INTERNATIONAL BUSINESS MACHINES PostScript is a registered trademark of Adobe Systems, INC

FOR MORE INFORMATION ON DYNA'S FULL LINE OF MOTHER BOARDS, SYSTEMS, FAX MACHINES, PostScript® SOFTWARE OR PERIPHERALS CALL 408-943-0100

CIRCLE 266 ON READER SERVICE CARD

Turn-key digital I/O.

- **48 programmable 8255 IC digital I/O ports**
- **6 Counter-timers (16 bit)**
- **Selectable interrupts**
- **Optional software in BASIC, C, and Turbo Pascal to save you *days* of development time**

the BEST

60-Day No-risk Return

D-MAX 54 P I/O Counter Board	$169
DIORES PLUS Software	$125

To order call
408-395-6600
FAX 408/354-4260

Ask for a FREE Catalog of Personal Computing Tools for Scientists and Engineers.

Personal Computing Tools, 17419 Farley Rd., Los Gatos, CA 95030

CIRCLE 268 ON READER SERVICE CARD

METRO COMPUTER CORPORATION

EVEREX 386/25, 4 MB, 44 MHD w/VGA	$7695
UBM 386/20, 1 MB, 44 MHD w/EGA MON	$3595
IBM/COMPAQ/AST/ALR	Call for pricing
KURTA IS/ONE, 12x12 w/single pen	$475
CALCOMP 23120, 12x12 w/4 button	$395
SUMMASKETCH, 12x18	$725
OTHER DIGITIZERS MODELS	Call for pricing
IOLP3700, A-E SIZE, 1 PEN	$3450
CALCOMP, 1043GT, 8 PEN	$6735
PHOTON 800 ..$705 **PHOTON 1000 +**	$995
NUMBER NINE PEPPER PRO - 1280	$2395

Call for what is not listed
1345 S. Anaheim Blvd.
Anaheim, CA 92805
TEL: 714-778-6495
FAX: 714-778-6877

CIRCLE 269 ON READER SERVICE CARD

DISTRIBUTORS & DEALERS WANTED

- For high performance SCSI host adapter.
- For Superfast ESDI to AT Bus 16 Bit controller with 15MB/s transfer rate

Contact ITD Corp.
Phone: 415-856-6686
FAX: 415-856-0819

CIRCLE 270 ON READER SERVICE CARD

Tech Marketplace . . .
Second Color Option Available

TECH MARKETPLACE

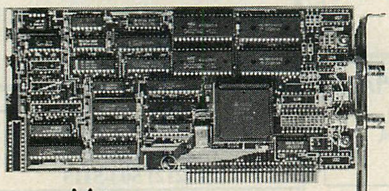
THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

COMMUNICATIONS

Turn your IBM PC into a 3270 Coax Data Analyzer.

FEATURES:
- Completely Passive
- Records up to 16 MB
- Function Keys, On-line Help
- Extensive Search Features
- Polls Compressed at Capture
- μ sec Accurate Time Stamps

OPTIONS:
- 3299 MPX, DFT, SNA
- Response Time Analyzer



Azure CoaxScope $2,995.00

14 Day Free Trial Evaluation

(617) 520-3800

TwinaxScope Also Available

AZURE TECHNOLOGIES
38 Pond Street, Franklin, Massachusetts 02038

CIRCLE 271 ON READER SERVICE CARD

COMPANION

Companion allows you to add an extra keyboard and monitor up to 250' from your PC system unit.

Operates with Mono, CGA and EGA monitors on IBM PC/XT/AT computers and 100% compatibles.

DEALER PROGRAM AVAILABLE.
PRICES START AT $219.00

CYBEX CORPORATION
2800 H. Bob Wallace Ave.
Huntsville, AL 35805
(205) 534-0011

CIRCLE 272 ON READER SERVICE CARD

INTERFACE CARDS for PC/AT and PS/2

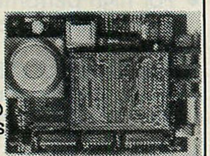
COMMUNICATION DATA ACQUISITION & CONTROL

FOR A FREE CATALOG CALL
1-800-553-1170

QUA TECH INCORPORATED
478 E. Exchange St., Akron, OH 44304
TEL: (216) 434-3154 FAX: (216) 434-1409
TLX: 5101012726

CIRCLE 274 ON READER SERVICE CARD

FAXBOARD $455
- 9600 BPS MODEM INCLUDED
- NO SEPERATE LINE NEEDED
- PAPERLESS SECURITY



BEYOND FAX $1177
SEND VOICE,DATA & COLOR GRAPHIC

SMART 8,16 PORT RS232 CARD FOR NETWORKING $700,$888.
& PS/2 MODEL 80 VERSION.
THESE ARE FEW OF OUR PRODUCTS
ASK FOR LATEST CATALOGUE

MAXTRON (818) 350-5707
1825A Durfee Ave., S. El Monte, CA 91733

CIRCLE 273 ON READER SERVICE CARD

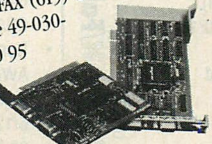
FREE 6' RS232 PC CABLE!

Recieve a high quality RS232 cable for your IBM PC. There is no charge or obligation. FREE! FREE! FREE! Just send your name, company name, address, and phone number to:

.... Softronics, Inc
.... 7899 Lexington Dr. Ste 210
.... Colorado Springs, CO 80920
.... Attn: Cables

Limit: 1 per company, while supplies last.

LAN Hardware
- New Technology Reduced Complexity & Cost
- Twisted pair–4 MBS Bus or Star
- Test Software Makes Installation a Snap
- Process Control, Industrial Net OS Software
- Drivers for Novell, All Netbios, Xinix Net
- Diskless PC, Chip level design
- OEM, VAR, Resellers only Call PC OFFICE OEM Div. (619) 268-3235, FAX (619) 268-0126 in Europe 49-030-323 30 95



CIRCLE 275 ON READER SERVICE CARD

AIS INC
421 E. Palatine Rd.
Palatine, IL 60067

312·359·2626

Software Link Inc:	
Multilink Advanced®	$399
Lanlink ™ 5.0 starter kit	$349
Lanlink ™ 5.0 satellite	$79
Lanlink™ 5X Server	195
Lanlink™ 5X satellite	$85
AT Gizmo™	$259
PC-Emulink™	$135
PC-MOS™ single user	$135
PC-MOS™ five user	$425
PC-MOS™ 25 user	$650

Digiboard serial boards:	
4S 4 port 16 bit I/O	$335
8S 8 port 16 bit I/O	$525
4I 4 port Intell I/O w/256k	$678
8I 8 port Intell I/O w/256k	$836

Call for PS-2 I/O boards

CIRCLE 276 ON READER SERVICE CARD

COMPUTER SYSTEMS

IBM PS/2's
Up to 41% off!

		Retail	Your Cost
MODEL 50-386	(New)		$Call
MODEL 60-041		5,295	3,395
MODEL 60-071		6,295	3,850
MODEL 80-041		6,995	4,395
MODEL 80-071		8,495	4,995
MODEL 80-111		10,995	7,895
MODEL 80-311		13,995	9,995

American Micro Computer Center
2890 Griffin Road - Suite # 4
Ft. Lauderdale, FL 33312

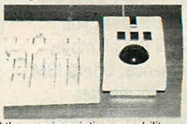
(305)985-9406

CIRCLE 277 ON READER SERVICE CARD

PERIPHERALS

FastTRAP™

The pointing device of the future is here!



- Two and three axis pointing capability
- High resolution trackball for X and Y axis input
- High resolution fingerwheel for Z axis input
- Use with IBM® PC's, XT's, AT's and compatibles
- Three input buttons
- Full hardware emulation of Microsoft® Mouse
- Standard RS-232 serial interface
- Includes graphics drivers and menu generator
- Easy installation
- 1 year warranty
- Made in the U.S.A.

ONLY $149.00
VISA and MasterCard accepted

LTS/C Corp.
319 South Limestone Street
Lexington, Kentucky 40508
(606) 233-4156
(800) 872-7279

3-D TRACKBALL FOR IBM

CIRCLE 278 ON READER SERVICE CARD

SPEECH SYNTHESIS

SynPhonix: TRUE Unlimited Speech Synthesizer for IBM-PC/XT/AT/jr & compatibles. This low power short card includes an SSi263 speech chip, amplifier and speaker. Software includes Text-to-Speech, Phonetic Editor, Talking Clock & demos. Can be programmed with BASIC and other languages. Prices start below $200.

SynPhonix™
Electronic Speech Articulator

Artic Technologies
55 Park St., Suite 2
Troy, MI 48083
(313) 588-7370

CREATE A DISKLESS PC!

PC-ROMDRIVE allows users to create a "Diskless PC" capable of booting a ROM-resident copy of MS-DOS and/or user application programs. PC-ROMDRIVE consists of a PC-compatible ROM/PROM expansion board and the PC-ROMDRIVE software. PC-ROMDRIVE is priced at $195 for single units. Quantity discounts and OEM arrangements available. MC/VISA

ALDIA SYSTEMS, Inc.
P.O. Box 37634
Phoenix, Az. 85069
(602) 866-1786

Tech Marketplace...

The comprehensive guide to products and services for the MS DOS market.

PERIPHERALS

Tech Marketplace...
Second Color Option
Available

Call (212) 503-5115 or
(800) 825-4ADS
For More Information

SOFTWARE

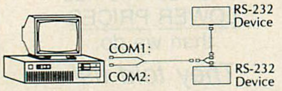
COMMUNICATIONS

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

COMMUNICATIONS

Communications Analyzer



- Monitors and stores to disk RS232 async data in real time
- Dual line DCE over DTE format
- Perform action on any data pattern
- Interactive mode
- Fully configurable, includes cables
- For IBM/PC/XT/AT or compatibles

Palladian Technologies Incorporated

$395.00
demo $10

1-800-537-5046 (301)576-0575

CIRCLE 283 ON READER SERVICE CARD

"E-mail...is a flexible bargain..."

—PC Week

Introducing SEAdog. An electronic mail system that's simple to set up, using standard phone equipment.

Faithfully send and retrieve electronic mail from one PC or from four hundred! And not just text, but spreadsheets, programs and database files as well. Per station $99.95

SEAdog

System Enhancement Associates
21 New Street
Wayne, New Jersey 07470
(201) 473-5153

CIRCLE 284 ON READER SERVICE CARD

ENGINEERING

DATA ENGINEERING GLOSSARY

Researchers, Educators, Engineers, Programmers, Managers, etc. Whether you are into Computers, Robotics, CIM, CAD/CAM: Now there is a glossary that can help. DATA ENGINEERING GLOSSARY over 5,000 entries and definitions. 3 disk package for just $29.95 plus $4 S/H. 5¼ or 3½ disks. Visa/MC accepted.
AAA DATA Engineering
P.O. Box 202
Sugar Grove, IL 60554-0202
(312) 897-6600 or FAX (312) 897-0024

GRAPHICS

FORTRAN Graphics Libraries

Now! 3 scientific graphics packages to support MS FORTRAN/Pascal, R-M or Lahey FORTRAN. GRAFMATIC (screen), PLOTMATIC (HPGL, H-I plotter) & PRINTMATIC (Laser & dot-matrix printer). All fully documented. Complete graphics primitives, 2-D plots, 3-D plots and solid models. $135 each, 2 for $240, all 3 only $340. Call or write for info.
Microcompatibles, Inc.
301 Prelude Drive Dept. J
Silver Spring, MD 20901
(301) 593-0683

35mm SLIDE FROM YOUR PC

COMPUTER SLIDE EXPRESS converts graphic files produced on the IBM PC into brilliant 35mm color slides with color resolution 400% better than your monitor. Leave your printouts behind. Use high resolution color slides up to 4000 line. COMPUTER SLIDE EXPRESS $9/slide.
VISUAL HORIZONS
180 Metro Park
Rochester, NY 14623
(716) 424-5300

LANGUAGES

THE SPINDRIFT LIBRARY

At last!! Now there's a way for the FORTRAN programmer to do the things they have always wanted to do: Execute other programs via CALL EXEC; direct control of the cursor with edit keys; WINDOWS on the screen for POP UP HELPs; (*) and (?) wildcard file searches; save/restore screen images; COLOR screens.

These are just a few of the features of the SPINDRIFT Library. Over 150 subroutines and functions in all, the SPINDRIFT Library includes DOS Interface (COPY, ERASE, MKDIR, FINDFILE, SYSTEM, etc), Variable Length Strings, Security Routines, Date/Time Routines, 10 SORT Routines, and much more!

Price: $149 plus shipping/handling
Write for a DEMO DISK: $5.00 credited toward purchase.
Specify your FORTRAN Compiler

Spindrift Laboratories, Ltd.
116 South Harvard Avenue
Arlington Heights, Illinois 60005
(312) 255-6909

CIRCLE 285 ON READER SERVICE CARD

BASIC to C

BASC_C v3.2 accepts BASICA, Quick BASIC, CBASIC86 and most other BASICs by adding the syntax & generation rule of new statements & functions (Customizer). Generates structured, indented, scoped MS/Turbo/Latice/Aztec C. All memory models and compile options are supported. C source (Runtime Library) incld. For MS/DOS, XENIX, UNIX. Demo disk. From $199.
GOTOLESS CONVERSION
P.O. Box 835910
Richardson, TX 75083
(214) 404-1404

LANGUAGES

ONLINE JCL!

Reference IBM OS and VS Job Control Language online via an interactive JCL Manual! Self paced instruction! Nearly 250 Keywords and Acronyms. 300 Pages and Screens! Basic and Advanced Techniques with Workshops. NEW! Two Diskette Drives or a Hard Disk required. IBM PC and Compatibles. $49.
DP Educational
6144 Sun Pattern Trail
Fairfax Station, VA 22039
Write for more info.

NETWORKING

KEYCARD ELIMINATOR™

NOVELL ADVANCED NETWARE® 2.0a & 2.1

KEYCARD FAILURE?
NEED TRUE SFT?
SLOTBOUND?
FAST CPU?

TM

Made in U.S.A.

$99.00 UPS Blue
TX orders add 8% Tax
CALL FOR DEALER PRICING

NBS DEALERS AND USER SITES INCLUDE:

BOEING	DREXEL BURNHAM	TENNECO
BORLAND INT'L	ENTRE	U.S. GOV'T
BUSINESSLAND	MICROAGE	UCLA
COMPAQ	MOBIL	UNIV. OF UTAH
CONOCO	OKIDATA	WESTERN DIGITAL

NBS

NETWORK BUSINESS SYSTEMS
1300 Woodhollow, Suite 5601, Houston, TX 77057
(713) 781-9268 (713) 783-4457

KeyCard Eliminator and the "no keycard" logo are trademarks of Network Business Systems
*NetWare is a Trademark of Novell, Inc.

CIRCLE 286 ON READER SERVICE CARD

NETWORKING

MP — THE NEW POWER OF NETWORK MULTI PROCESSING

MULTI PROCESSING DEVELOPMENT ENVIRONMENT FOR PC LOCAL AREA NETWORKS.

"stop multi-tasking, start multi-processing"

- **MP** is a state of the art multi processing environment for MS-DOS 3.10 or higher and PC local area network supporting NetBIOS protocol or NetBIOS emulator.
  MP is a software toolbox made of several packages, opening a wide door to programmers for real multi processing and distributed applications.
  At present, Microsoft C 5.00 compiler interface is available, soon Microsoft Pascal, Basic and Fortran interfaces will be available.

MP PACKAGES AND PRICES

- **MP STARTUP** – Software Toolbox enabling "master" to submit a process to any available or dedicated network station "slave" for remote execution. Remote execution can be synchronized and controlled by master using named pipe duplex communication protocol. Named pipe interface is simple and easy to use as the MS DOS file system interface ("pipe open", "pipe read", "pipe write". . .).
  Object price: $99.00
  Source price: $425.00
- **MP SERVER** – Software Toolbox enabling to create any purpose Network Server. With MP SERVER programmer can create servers as communication server, file server or any other server that can simultaneously support up to 100 stations in a single network.
  Object price: $445.00 (including interface source)
  Source price: $2025.00 (including all source)
- **MP MASTER** – End user utility, used for submitting applications to remote available slave while remote execution (e.g. compilation, printing etc. . .) user can use local station for other tasks (e.g. word processing, editing etc. . .).
  Object price: $49.00
  Source price: $325.00
- **MP NETBIOS** – low level interface to NetBIOS (that can be easily ported to another environment) including all Network and NetBIOS routines with station multitasking options.
  Object price: $49.00
  Source price: $125.00

(Add $20 for S/H)

TO ORDER OR FOR ADDITIONAL INFORMATION CALL
TEL: (972) (53) 333931, (972) (53) 621905 FAX: (972) (53) 342418
OR WRITE TO

SHANY

SHANY COMPUTERS LTD
Rechter Building, 4 Smilansky Street, Natanya, ISRAEL 42432

CIRCLE 287 ON READER SERVICE CARD

PC TECH JOURNAL MARKETPLACE is a special economical section for product and service listings.

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

NETWORKING

NETWORK CONTROL LIBRARIES
NETWORK INTERFACE allows file sharing and redirection through DOS functions. $99. NETBIOS ROUTINES allows access to low-level network functions. Name, session & datagram routines. Wait and no-wait options. $199. NETWORK MASTER provides access to Netware internal functions. $299. Complete network control from your compiled programs!

..........Starlight Software..........
..........P.O. Box 1090..........
..........Wheeling, IL 60090..........
..........(312) 394-0622..........

MULTI PROCESSING SORT
MP SORT - Simultaneously uses several Network slaves for achieving high performance sorting of any file or named pipe information (To use MP SORT with named pipe, MP START-UP is also needed). Object price: $99.00. Source price: $325.00 (Add $20.00 S/H).
SHANY COMPUTERS LTD
4 Hahimeir Street
Natanya, Israel 42304
TEL: (972) (53)621905 FAX: (972) (53)621905

PROGRAMMERS TOOLS

BIOS SOURCE CODE

The **XT BiosKit** is a 270 page book with a diskette containing source code in C, plus utility programs to help you create a Bios. Now you can have a Bios with documentation for your own applications: modify boot-up, eliminate the keyboard, install security features, etc. Only $99 complete. The **AT BiosKit** is only $199, or get both Bioskits for $279.

— XT-AT HANDBOOK —

The **XT-AT Handbook** is full of hardware and software information in a shirt pocket size book. Over 70 pages covering 38 subjects, including connectors, I/O maps, controller programming, DOS and DEBUG commands, board dimensions, character codes, hard disk drive types, and much more. Only $9.95 each qty 1-4, five or more, $5 each.

Annabooks
12145 Alta Carmel Ct. 250, Suite 262
San Diego, California 92128
(619) 271-9526
CIRCLE 288 ON READER SERVICE CARD

Attn: Systems Developers . . .

Add-a-Terminal®

Adds exact terminal emulation

. . . dynamically link exact terminal emulation to your existing software. Every keystroke, every video presentation, every printer function is reproduced *exactly.* Written in assembler, it conserves memory and delivers unparalleled performance.

Add Terminal Emulation to:
- Communications Software
- TCP/IP & X.25 Gateways, etc.
- LAN Environments
- INT 14 Communications Redirectors
- Custom Applications

Migrate your host software to micros . . .
. . . without modifying keyboard and screen handling code. Application programs read the PC keyboard and write the PC screen, through Add-a-terminal®, just like the real terminal.

Over 50 Terminal Emulations to choose!
Including: DEC, Data General, ADDS, Datapoint, Hazeltine, Hewlett-Packard, Honeywell, IBM, Lear-Siegler, Perkin-Elmer, Prime, Televideo, TI, Wyse, and More.
Custom emulations available.

ECONOMICAL VOLUME & BUNDLING PRICING AVAILABLE!

1-800-225-8590

SOFTRONICS 719/593-9540 FAX: 719/548-1878 TELEX: 450236
CIRCLE 290 ON READER SERVICE CARD

C & MASM — DOS & OS/2

VersiMAKE™
A full-featured MAKE utility that derives your system's dependencies through analysis of your C & MASM source files. No more MAKE dependency files to maintain!

VersiCREF™
A unique utility that creates a sorted Master Cross-Reference of your entire system. Handles 100+ C and MASM source files. Full X-Ref or just PUBLIC symbols.

VersiMAKE™	$125
VersiCREF™	$75
Both	$150

800-334-4096
(In NJ, 609-871-0202)
MC/VISA/AMEX

Free Demo Disk

SUMMIT INFORMATION SYSTEMS, INC.
73 East Lane, Willingboro, NJ 08046
CIRCLE 291 ON READER SERVICE CARD

PROGRAMMERS TOOLS

TRANSLATE BASIC TO PASCAL

P-TRAL, will convert your MS BASIC programs to Turbo Pascal. It reads the BASIC source code to produce Pascal source. It steps you through the translation process. "*...easier to use than other BASIC-to-Pascal translators...the code P-Tral generates is somewhat cleaner than others...*"
Micro Systems Journal, June 1988

For $179.00, it comes with complete documentation and tutorial. Requires MS or PC- DOS version 2.0 or later.

WOODCHUCK INDUSTRIES, INC.
340 WEST 17 STREET Ste 2B
NY, NY 10011 • (212) 206-6490/924-0576
CIRCLE 292 ON READER SERVICE CARD

C-ARMOR™ TOOLKIT

Bullet-Proof MULTI-TASKING C Programmer's Development Library! Supports:
- Mouse
- DOS
- Menus
- Lists
- Forms
- Error Handling
- TimeSlicer™!

C-ARMOR™ LIBRARY $150, C-ARMOR™ Source ADD $100, Forms ADD $100, TimesSlicer™ ADD $250. FREE Optical Mouse with COMPLETE ORDERS. FREE DEMO available. Dealers Welcome! CASH/COD. FREE NEXT-DAY for Pre-Paid Orders. OS/2 Version coming SOON!
TimeSlicer is a trademark of TimeSlice, Inc.
C-ARMOR is a trademark of L² SOFTWARE COMPANY

L² SOFTWARE COMPANY
P.O. Box 4235
Englewood, CO 80155-4235
To Order: **(800) 950-55L2**
For Support: **303-771-3190**
CIRCLE 293 ON READER SERVICE CARD

Get Inside Your Program Today!

Analyze and improve your program . . . function by function! Inside follows your program through execution counts, minimum, maximum, and total elapsed times with microsecond accuracy!

Inside! Turbo C
Inside! Turbo Pascal
Inside! Quick C
Inside! Quick Basic
Inside! Microsoft Pascal
Inside! Microsoft Fortran
Inside! Lattice C
Inside! Logitech Modula-2

$75.00 Each

Visa/Mastercard Accepted
(800)537-5043
Paradigm Systems
P.O. Box 152 Milford, MA 01757
CIRCLE 294 ON READER SERVICE CARD

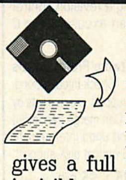
ROM DEVELOPMENT TOOLS

Link MS-DOS and Intel object files to ROMable code. GeneLink permits independent placement of each program segment for complete flexibility in memory layout. GeneLink links directly from object files to HEX or other formats, and is five to ten times faster than other locating tools.

Genesis has been a major supplier of ROM development tools to major industrial customers like IBM and Boeing for over five years. You can count on Genesis to provide reliable, fast tools and excellent support.

Genesis Microsystems
Genesis Microsystems Corporation
13300 Sunset Drive, Los Altos, CA 94022
Call: (415) 964-9001; TX: 4998093 GENMS UI
CIRCLE 289 ON READER SERVICE CARD

REVERSE ENGINEERING TOOL



Force software to reveal secrets! INTERCEPTER/RECORDER records requests to DOS and BIOS and gives a full report. Completely invisible.
Only $34.95 post paid.

JACKSON SOFTWARE
P.O. Box 9317, M.S. 10, Baltimore MD 21228
CIRCLE 373 ON READER SERVICE CARD

'C' DOCUMENTATION TOOLS

- * **C-CALL** ($39) creates a graphic-tree of the caller/called structures, and creates a files-vs-procedures table of contents.
- * **C-HDR** ($39) create/insert/update header for each procedure showing caller/called and local/global/parameter identifiers.
- * **C-LIST** ($29) lists and action-diagrams, or reformats (incl. comment alignment).
- * **C-REF** ($29) gives cross-reference of local/global/parameter identifiers.
- * **SPECIAL!!** All for $99 (30-day guarantee) plus FREE integrated **C-DOC** program.

SOFTWARE BLACKSMITHS INC.
6064 St Ives Way,
Mississauga, ONT, Canada. L5N-4M1
(416) 858-4466
CIRCLE 374 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PROGRAMMERS TOOLS

The Developer's Data Manager

AXOS™

It Speaks Your Language!

BASIC • COBOL • C • FORTRAN • PASCAL • ASSEMBLER

A New Standard in Access Method Technology

AXOS is a *fast* full-featured, key-indexed file management system designed to provide application programmers and software developers a flexible and versatile data storage and retrieval mechanism with the least amount of overhead. It can be used in either single or multi-user LAN environments on the IBM PC PS/2 and compatibles. As AXOS can be called from any programming language through a common communication protocol, it allows one to manipulate data in a manner not possible with other DBMSs or access methods. AXOS also offers a full complement of file management functions, system management functions, productivity tools and debugging aids designed to ensure easy and accurate development and support of microcomputer applications systems.

FEATURES

- SECURITY: LAN Computer/password and File I/O funtions by computer
- AUDIT: Complete or partial capture of all AXOS activity, available online or hardcopy
- DATA INTEGRITY: Ensures full data integrity in the event of a program, system or power failure
- INTERACTIVE MONITOR: Instantaneous interactive monitoring of AXOS activity as a program is executing
- FILE CONVERSION: Converts many types of files to AXOS format
- FILE LIST: Produces a formatted report of AXOS file contents
- RECORD ENCRYPTION AND COMPRESSION
- PROGRAM INVOKED ERROR ROUTINE: For processing unexpected I/O return codes

SPECIFICATIONS

- Languages Supported
  BASIC COMPILED BASIC COBOL
  FORTRAN C ASSEMBLY PASCAL
- Local Area Networks Supported
  IBM® PC NET AT&T® StarLAN
  3Com® 3+Share Novell® Network
- File Types Supported ..: Sequential
  Single Index Multi Index (cluster)
- Maximum Indexes ....: 64 Per File
- Maximum Key Length : 240 bytes unique
  ....: 236 non-unique
- Maximum Record Size : 64,000 bytes
- Maximum Records
  Per File ....: 16,777,216
- File Capacity ....: 4.2 billion bytes
- Maximum Files & Indexes allowed
  open at one time ....: 253
- Operating System Requirement:
  DOS 2.1 or above; 128 - 256KB of memory

Come To Grips With The Power Of AXOS

McSoft INCORPORATED™

$595 Includes 30 days free technical support. Extended technical support available

And there are no royalties on your AXOS applications!

TO ORDER CALL (404) 237-4169 AMEX/COD/MC/VISA

PO Box 12044 • Atlanta, GA 30355-2044

CIRCLE 296 ON READER SERVICE CARD

ICON-TOOLS™ .... from Ithaca Street Software, Inc.

Powerful, full-featured editor. Icon capture program-clip icons from .IMG, .PCX, art files. Function library in C with full source code. Use with Turbo C BGI graphics, Quick C graphics, MetaWINDOWS, Essential Graphics, GFX. Only $69.95, and worth every cent! Money-back guarantee.

1145 Ithaca Drive, Boulder, CO 80303 (303) 494-8865

CIRCLE 299 ON READER SERVICE CARD

SOURCER™

Commenting Disassembler

Creates detailed commented source code and listings from memory, .COM files or .EXE files directly suitable for assembly. Built in data analyzer and simulator resolves multiple data segments and provides detailed comments on interrupts and subfunctions, I/O ports and much more. Outstanding review in *PC Magazine* 4/26/88 page 46.

BIOS SOURCE

PS/2 ■ AT ■ XT ■ PC ■ Clones

The BIOS Pre-Processor to SOURCER™ provides the first means to obtain accurate legal source listings for any bios! Identifies entry points with full explanations. Provides highly descriptive data labels such as "video_mode" and much more. Fully Automatic.

SOURCER	$ 99.95
SOURCER w/BIOS Pre-Processor	$139.95

(Shipping $3, outside USA $15, CA Res. add tax)

To order or receive information just call!

1-800-662-8266

V COMMUNICATIONS
3031 Tisch Way, Suite 905, Dept. T
San Jose, CA 95128 • (408) 296-4224

PS/2, AT, XT, PC are trademarks of IBM Corp.

CIRCLE 300 ON READER SERVICE CARD

ROM YOUR ".EXE"

EXELOC v3.0 creates rommable files for 8088/8086 based systems from MS-DOS ".EXE" files.

Version 3.0 allows full control of segment locations.

Supports initialized data segments.

Creates binary or INTEL HEX-ASCII files.

Run EXELOC in batch or menu mode.

EXELOC is what you need for:

CASH REGISTERS
ROBOTICS
INDUSTRIAL CONTROLLERS
BIOS DEVELOPMENT
DISKLESS WORKSTATIONS
. . . OR ANY FIRMWARE

Very fast.

Only **$79.95** U.S. plus $3 s/h. CHK/MO.

VIRTUAL SOFTWARE
150 Alton Towers Circle, Suite 605
Scarborough, Ontario, Canada
(416) 754-0711

CIRCLE 365 ON READER SERVICE CARD

ARRAY SUBROUTINES

Faster development and execution, 174 assembled routines for interpreted or compiled BASIC, Assembler, Pascal. Sort, grade, resequence, expand, locate, compare, etc for floating point, integer, and string variables; and add, and, rotate, shift, xor, etc for logical (binary) vectors, $49.99.

..........uniSON Software..........
..........Box 2033..........
......Moose Jaw, Sask. S6H-7N7......
........Canada, (306)-694-0072........

ROM Microsoft-C $495

One integrated package for ROM development with Microsoft-C®. The package has full Source level remote debugger (locals, register variables), 8086 locator that produces Intel hex, Absolute OMF (with locals and Basic types), Startup code and a ROMable library (printf).

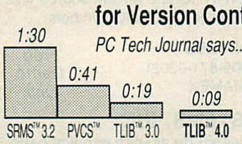
Call us for complete information.

Datalight
17505 - 68th Ave. N.E. #304
Bothell, WA 98011 USA
(206) 486-8086
1 (800) 221-6630

---

for Version Control

PC Tech Journal says...



Times are to update a 45K library on a PC/XT. PVCS and TLIB 3.0 are from Sept 87 *PC Tech Journal.* SRMS and TLIB 4.0 are later versions.

TLIB™ is *FASTEST!*

"TLIB is a great system" ***PC Tech Journal, March 88***

"TLIB...has my highest recommendation." *Ronny Richardson,* ***Computer Shopper, August 87***

"If you've been putting off getting a revision control system, you no longer have an excuse." ***The C Users Journal, February 88***

- **A Full-Featured System for Software Professionals**
  Branching, for parallel development. Check-in/out locking. Keywords. Wildcard and list-of-file support; creates lists by scanning source code for includes. Can merge (reconcile) multiple simultaneous changes and undo intermediate revisions. Network and IBM 3363 optical disk support.

MS-DOS 2.x & 3.x ***Just $99.95 + $5 s/h*** Visa/MC
5 station LAN license $299.95 + $5 s/h
call for pricing on other network sizes

BURTON SYSTEMS SOFTWARE
PO Box 4156, Cary, NC 27519 **(919) 856-0475**

CIRCLE 298 ON READER SERVICE CARD

OPT-TECH SORT/MERGE™

The *High Performance, High Quality,* Sort/Merge utility. Use as a stand-alone routine or Call as a subroutine to over 30 languages. Unlimited filesize, multiple keys, record selection, and much more! MS-DOS $149.

To Order or for more Information, Call or Write.

Opt-Tech Data Processing (702) 588-3737
P.O. Box 678/Zephyr Cove, NV 89448

CIRCLE 366 ON READER SERVICE CARD

ATTENTION ANALYSTS & DESIGNERS

Introducing SpecWriter™

Now you can use ***SpecWriter*** *to design system components utilizing*
• *Screen Painting* • *Screen Prototyping* • *Report Painting*
• *Report Prototyping* • *Layout Development* • *Word Processing*

Create complete Functional Specifications easily using this innovative, non-procedural micro package. Supports variable sized screens and reports. MICRO and 3270 types

ORDER TODAY OR CALL FOR YOUR FREE DEMO DISK

TOLL FREE ***1-800-541-2071*** *(Outside California)* • ***1-619-560-1414*** *(Inside California)*

$345 + $5 Shipping & Handling VISA-MC-Checks Accepted
Accelerated Systems, Inc. • *4951 Clairemont Square, Suite 298, San Diego, CA 92117*

CIRCLE 367 ON READER SERVICE CARD

Tech Marketplace Home of the Power Buyer

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PROGRAMMERS TOOLS

SCREEN MANAGER

MENU, WINDOW, and DATA ENTRY Support for the Professional Programmer! Interfaces to most languages. BASIC, C, FORTRAN, COBOL, PASCAL, ASSEMBLER. 100 Page Manual. Thirty day money back guarantee. No Royalties.

from **$79** VISA/MC

The West Chester Group
P.O. Box 1304
West Chester, Pa 19380
(215) 644-4206

CALL FOR FREE DEMO

CIRCLE 368 ON READER SERVICE CARD

FREE BUYER'S GUIDE

Programmer's Connection is an independent dealer representing more than 300 manufacturers with over 800 software products for IBM personal computers and compatibles. We have serviced the professional programmer since 1984 by offering sound advice and low discount prices. Call or write today to receive your **FREE** comprehensive Buyer's Guide.

US	**800-336-1166**
Canada	800-225-1166
International	216-494-3781

Programmer's Connection
7249 Whipple Ave Nw, N. Canton, OH 44720

CIRCLE 369 ON READER SERVICE CARD

One-stop ROM Shop

When your application calls for ROM support, contact the professionals at ALDIA SYSTEMS. We've been supporting firmware developers for over two years with excellent software development tools and technical support. Don't settle for imitations, call the company who brought you ***PC-LOCATE***, the original PC locater.

PC-LOCATE: Produce ROM-able code from your ".EXE" files. PC-LOCATE assigns physical addresses to the re-locatable image based on user inputs. PC-LOCATE support the entire Intel processor family including : 8086, 8088, 80186, 80188 and 80286.

PC-PROMPAK: A PROM/ROM expansion board for IBM and IBM-compatible computers. PC-PROMPAK provides up to 384Kbytes of non-volatile expansion memory and supports most 28-pin JEDEC devices including EPROMS, EEPROMS and Static RAMs.

PC-ROMDRIVE: Create a "Diskless PC" that can include MS-DOS and your application program. "Autoexec.bat" files are supported for automatic program execution.

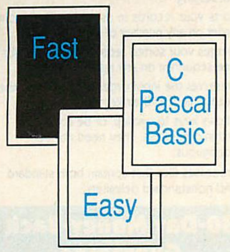
ALDIA SYSTEMS, Inc.
P.O.Box 37634 Phoenix, Az. 85069
(602) 866-1786

CIRCLE 371 ON READER SERVICE CARD

LyBTREE—B-tree TOOLBOX

A compact and extensively tested algorithm. LyBtree provides all necessary documentation and examples. Easy, high level interface with all required libraries for a quick and seamless installation. $150 source code included. Manual only $20, REFUNDABLE. MC/VISA.
Labyrinth Systems, Inc.
30 Buxton Farm Rd.
Stamford, CT 06905
(203) 322-5800

HIGH PERFORMANCE WINDOWS



SCREEN MANAGER
PROFESSIONAL

- All source code included (100% machine code)
- Unlimited number of windows
- EGA. CGA. MGA support
- Write using direct video or BIOS access
- Fuzz and Flicker eliminated
- Drag windows around the screen
- Full support for overlapped windows
- Pull-down and Lotus type menus
- Royalty free!
- Many more features packed in over 80 functions

The **Screen Manager Professional** is a state-of-the-art windowing facility. Save time, money, and aggravation without a sacrifice in quality. Powerful enough for the toughest application yet so easy to use that now even beginning programmers can create clean, professional looking programs.

Amazing Speed! Incredible Price! **$129.95**

LAI LOGICAL ALTERNATIVES INCORPORATED
PO BOX 10674, State College, PA 16805
(814) 234-8088 *Demo Available*

CIRCLE 370 ON READER SERVICE CARD

PROGRAMMERS TOOLS

THE MISSING LINKer

FIRMWARE DEVELOPMENT TOOLS for
MICROSOFT® C

MICROSOFT® C

LINK & LOCATE++
SUPPORTS iAPX 86/87/186

Downloads to IN-Circuit Emulators with INTEL™ OMF or SoftProbe™ II Target Debugger

ROMABLE CODE

Includes...
- Start Up Files
- Linker
- Locator

- Library Support with Floating Point Operation
- COMPLETE Microsoft™ C Debugging Information

All This PLUS Other Fine Products from...

SYSTEMS & SOFTWARE, Inc.

3303 Harbor Blvd. • C-11
Costa Mesa, CA 92626
714/241-8650

FREE
PRODUCT CATALOG AND DIGEST...
WRITING ROMABLE CODE USING MICROSOFT C

CIRCLE 372 ON READER SERVICE CARD

PROTEUS: PROTOTYPE/DEMO SYSTEM

NEW VERSION! $149.00

Proteus 4.0 now supports data entry fields, branch on match and range, video effects, custom sound and much more. Export Proteus screens and forms to popular interface management systems. Still with the best text drawing package on the market and a friendly Borland-style interface. Find out why designers agree Proteus is the system that's "Better than Bricklin."

Phone orders: (800) 634-9986
No royalties
30-day Money Back Guarantee
MC/VISA/COD/PO accepted.

Helios

Helios Software • P.O. Box 22869 • Seattle, WA 98122

CIRCLE 295 ON READER SERVICE CARD

HyperDraw for the IBM PC

ATTENTION PROGRAMMERS! Create database stacks for the PC family using the new HyperDraw features of PC-KEY DRAW Version 3.5. Use powerful graphics editor with freeform Hypertext for creating the databases of the future. Hundreds of applications. Only $100 for full registered version.
OEDWARE
P.O. BOX 395
Columbia, MD 21045-0595
(301) 997-9333

—OS/2 TOOLS-OS/2 TOOLS—

Save precious time and effort on your OS/2 and DOS development. Our Libraries contain Utilities including: file, screen, keyboard, window, string functions and more. Includes OS/2 header files! All fully documented and supported. Also great Screen and Menu editor $39. Libs: OS/2 version $99; DOS version $49. MC VISA.
OACIS INTERNATIONAL, INC.
1300 N.W. 99 Ave
Plantation Fl. 33322
(407) 997-6472

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PROGRAMMERS TOOLS

BTRIEVE™ BTRIEVE™ BTRIEVE™
If you use Btrieve, you must have BUTILITY. BUTILITY is the only way to edit and create Btrieve files (ver 4.0 and up). Stop wasting time writing those utility programs! Use QB 4.0 look and feel editor to perform all operations, edit the data buffer, edit actual key values, search and replace, and much, much more. Only $89.95 + $5 S&H. MC VISA or COD.
SMITHWARE, Inc.
3600 Hillsboro Pike Suite B-13
Nashville, TN 37215
(615) 297-4876

CALLABLE WORD PROCESSOR
A Word processing capabilities are now available from within applications. By specifying a row and column on the screen, developers can edit files from within any environment (C, dBase, Pascal, etc.). Features include, cut/paste, word wrap, margins, and word search. $59. Add $20 for C source.
Wilson Software
3600 Dorshire Court
Pasadena, MD 21122
(301) 255-5717

TOOLS FOR MS WINDOWS & PRESENTATION MANAGER
For accounting or other networked multifile applications. Speed development! Save months! Four modules: graphical file maintenance, start up (password etc.), help and report. Must have SDK. Source available. No royalties. MW Tools $995, PM Tools $1495.
The Software Factory
Gainsville, FL
904-376-8370

SECURITY DEVICES

EVERTRAK

SECURITY

If you HATE Copy Protection, but still want to protect your program from hackers and reverse engineering, EVERTRAK allows you to:

- Shut down DEBUG systems and disassemblers.
- Restrict use by date-range and media-type.
- Embed a secure serial number string.
- Track unauthorized copies back to their source.

EVERTRAK is totally invisible to your end user and uses the latest encryption technology.

EVERTRAK supports IBM & Compatibles using DOS 2.0 or later, sells for $295.00 and has a 30 day money back guarantee.

AZ
Az-Tech Software, Inc.
305 East Franklin
Richmond, MO 64085
(800) 227-0644 (816) 776-2700
CIRCLE 376 ON READER SERVICE CARD

SECURITY DEVICES

COPY PROTECTION

The world's leading software manufacturers depend on Softguard copy protection systems. Your FREE DISKETTE introduces you to SuperLock™ – invisible copy protection for IBM-PC (and compatibles) and Macintosh.

- Hard disk support
- No source code changes
- Customized versions
- LAN support
- New upgrades available

(408) 773-9680

SOFTGUARD SYSTEMS INC.
710 Lakeway, Suite 200
Sunnyvale, California 94086
FAX (408) 773-1405

CIRCLE 375 ON READER SERVICE CARD

SOFTGUARD

EVERLOCK

COPY PROTECTION

Designed for user-transparency, clone compatibility & strength. It features:

- no need for damaged media or I/O plugs
- supports all Hard & Floppy disk formats
- file-server networks supported
- variable number of installs (0-99)
- create demos with remote unlock option
- allows protected upgrades by modem/BBS
- limit program use by date or count

$195 Starter Kit or $495 with NO meter counts. Free info & demo disk available.

AZ
Az-Tech Software, Inc.
305 East Franklin
Richmond, MO 64085
(800) 227-0644 (816) 776-2700
CIRCLE 377 ON READER SERVICE CARD

BIT-LOCK® SECURITY
Piracy SURVIVAL 5 YRS proves effectiveness of powerful multilayered security. Rapid decryption algorithms. Reliable security device transparent to PARALLEL or SERIAL port. Complemented by low cost KEY LOK™ & multifeatured COMPU-LOCK™. Including countdown, timeout, data encryption & multiproduct protection.
MICROCOMPUTER APPLICATIONS
3167 East Otero Circle
Littleton, CO 80122
(303) 922-6410 or 770-1863

To place your ad in **Tech Marketplace**
Call (212) 503-5115 or **(800) 825-4ADS**

SOFTWARE PACKAGING

Software Packaging & Computer Supplies

• IBM style binders & slip cases - blank or printed • Vinyl binders & slip cases • Vinyl folders & packages • Continuous & cut sheet paper for manuals 5-1/2 x 8-1/2 • Disk pages • Disk envelopes • Continuous & laser labels • Shipping cartons & envelopes for software & MORE !!

Call For A Free Catalog!!
800-DEAL-NOW 312-629-5160
Anthropomorphic Systems, Ltd.
376-J East Saint Charles Road
Lombard, IL 60148
CIRCLE 378 ON READER SERVICE CARD

TERMINAL EMULATION

BARR High-Performance RJE Workstations

BARR/SNA RJE and BARR/HASP are communications hardware/software packages for the IBM PC and PS/2 series of computers.

Full featured:

- multiple printers,
- unattended operation,
- print speed beyond 6,000 lpm,
- special forms,
- line speed to 56,000 bps,
- dial-up or dedicated lines, and
- serve as a LAN gateway.

BARR/SNA RJE emulates IBM 3777-3 in an SNA environment. BARR/HASP emulates IBM 3777-2 and HASP on the 360/20. BARR communications software is also available with PC-SYNC internal modems: 208AB, 201C, 9600.

BARR
Barr Systems, Inc.
2830 NW 41 St. Bldg M
Gainesville, FL 32606
800-BARRSYS or
904-371-3050
CIRCLE 379 ON READER SERVICE CARD

UTILITIES

1000 dBASE Utilities!!!!
Massive library of all PD/Shareware utils for dBase/Compiler programmers. Assembled from national BBSs and private collections in the US and Europe, this incredible collection has >500 utilities on 25 + disks in ARC format. Includes index DBF of these and >500 commercial utilities. $99.95, VISA, MC, POs & COD.
EMS
11701 Karen Drive
Potomac, MD 20854
CALL 301-299-9239 for more information

UTILITIES

If you can find better sort/merge software, buy it!

Sortex:® A flexible, general purpose sort/merge facility which provides you with the ultimate in performance and reliability

- Runs from the DOS prompt, batch file, or applications written in Assembler, BASIC, COBOL, FORTRAN, Pascal, or C.
- Accepts your requirements through the full screen menu/panel facility, control file, or command line.
- Processes most of the commonly used file and data types.
- Produces output files with key, address, index, tag, or full records.
- Imposes no practical limitations. Files with four billion characters, records & fields with thirty two thousand characters each, and unlimited number of keys can be handled by Sortex.
- Allows records to be chosen selectively for processing.
- Sorts your records in ascending/descending order on any number of keys.
- Merges your sorted record sequences into one sequence on any number of keys.
- Preserves the input order of records on the output file if all their keys are identical.
- Allows keys to overlap, or be contained within other keys. They need not be contiguous.
- Processes files that contain both standard and nonstandard delimiters.

30-Day Money-Back Guarantee
Only $149.95

Systemat
CORPORATION
1915 W. Orangewood Avenue
Orange, California 92668
Tel.: 714 978 0480
CIRCLE 380 ON READER SERVICE CARD

UTILITIES Category
continued on next page

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

Utilities

TOKEN-RING CONTROL

PRODUCTIVITY TOOLS is indispensable for network and workstation architects. Proven in hundreds of installations, this package combines all the tools you, and your users, will ever need.

- Extensive menuing with context -sensitive help (resident and non-resident options).
- Full control of software distribution and updates.
- Intelligent, automated backup of any combination of files and directories on any workstation.
- Interactive and programmable file management for locating, moving and undeleting files.
- Command line editor with synonyms, recall of all previous commands plus full screen editing.
- Temporary connections to shared devices plus forced release of print jobs.

ONLY $99.

Site licenses available

Applied Automated Engineering Corp.
65 S. Main St. Pennington, NJ 08534
(609) 737-6800

CIRCLE 381 ON READER SERVICE CARD

DISK ACCELERATOR V2.0

DiskCache speeds up your hard disk access. Disk caching and ram disk in one package. Ram disk shares cache space. Transparent, flexible, configurable, no h/w changes. RAM, EMS, and AT extended memory versions incl. Not copy protected. VISA, MC, volume discounts. No PO's w/o prior approval. $49.00
DATAMORPHICS LTD.
P.O. Box 820
Stittsville, Ontario, Canada K0A 3G0
Or call (613) 831-0409

SAVE DEBUGGING TIME!

WATCH-I/O is a popup utility that shows names and technical info of all files in use. WATCH-I/O also shows name of last file opened, current program name, and last file-not-found. Only $49.95 + $5 S&H Visa/MC/CK/MO.
Quantum Publishing
23860 Miles Road
Cleveland, OH 44128
1-800-342-6237

COMPRESS YOUR DATA!

ARC is considered the industry standard archive utility. It's the file storage method used by PCTECHline. And PC WEEK called it "A sophisticated and eminently useful product."

ARC automatically compresses stored data so it takes up less space. And less modem transfer time. From 20% to 90% less, depending on the kind of data.

All program sources are included and it's not copy protected .... **$50 ARC**

sea System Enhancement Associates
21 New Street, Wayne, New Jersey 07470•(201) 473-5153

CIRCLE 383 ON READER SERVICE CARD

AT's DON'T NEED 360KB DRIVES

Allows copying of IBM AT files for use on IBM PCs, XTs, and compatibles with no hardware or software modifications. A 360K floppy drive is not required. "CPYAT2PC" may reside on your IBM AT hard disk and copies 1 file or entire subdirectories in 1 step. Price $79. + $4 s/h

MICROBRIDGE COMPUTERS INTERNATIONAL, INC.
655 Sky Way - Suite 125
San Carlos, CA 94070
1-800-523-8777
(CA) 415-593-8777
(NY) 212-334-1858

CIRCLE 384 ON READER SERVICE CARD

A BEAUTY FOR YOUR BEAST

STILL RIVER SHELL, NEW VER 2.44. Faster more powerful version of this popular DOS shell. Find copy, move, delete files in a few keystrokes. *Tagging *Find on name and text *Full Tree *User defined commands and more. Unltd. site lic. $390. $39 complete with 160pp. manual. MC/VISA. 90-Day No-Risk Guarantee.
Bill White
P.O. Box 57
Still River, MA 01467
(508) 456-3699

INSTANT MENUS

Create colorful, detailed "pull down" style menus for any program... WITHOUT PROGRAMMING! Works with or without a mouse. Comes with menus for Word Perfect, Procomm, MS-DOS and more. Makes software easier to use and minimizes training time by creating the perfect user interface. $65.
Pliable Products
420 East Broad Street
Souderton, PA 18964
(215) 721-7283

Fix common problems fast!

You don't need to be an expert to diagnose and correct problems involving PC setup. All you need is HELPME™ software! More than 300 tests. Quick identification of system configuration and compatibility. $99 + S/H. MC/VISA. California Software Products, Inc., 525 N. Cabrillo Park Drive, Santa Ana, CA 92701, (714) 973-0440.

CIRCLE 382 ON READER SERVICE CARD

Utilities

PS/2 SERIAL I/O DRIVER

COM1-COM8 DOS device driver for PS/2 Models 50, 60, and 80. Interrupt driven, buffered I/O. Loaded via CONFIG.SYS. Extended BIOS for buffer control and monitor. All serial lines accessible via high level languages or BIOS calls. $129.00 + $3.00 S/H.
Diamond Control Systems, Inc.
400 W. Bethany Drive, Suite 140
Allen, TX 75002
(214) 727-8511

Miscellaneous

Bar Coding

BAR CODES MADE EASY

PERCON® E-Z-READER™

FAST • ACCURATE • RUGGED

NO SOFTWARE CHANGES with PC/XT/AT and PS/2, AT&T 6300, Wyse 30/60/85/PC/AT, Kimtron KT-7/PC, Link PCTerm/MC1/MC3, TeleVideo PCS1/905/955 & DEC VT220 keyboards, multi-user RS-232 interfaces.

- Immediate shipment
- Free phone support
- 2-year warranty

Details or Questions? Call us.
(503) 344-1189
2190 W. 11th Ave., Eugene, OR 97402

See us at COMDEX #S9722

PERCON®

VARIANT MICROSYSTEMS

BAR CODE READERS DELIVER

WAND/ LASER/ MAGNETIC CARD CONECTIVITY

- Keyboard wedges (Internal/External) for IBM PC/XT/AT,PS/2 and portables.
- RS232 wedges for WYSE, Link, Kimtron terminals
- Autodiscriminate between codes 3 of 9. UPC, EAN,12 OF 5. Codeabar. Code 128
- Interface with wands. Symbol Technologies' LS 7000/8000 and Photographic Sciences lasers
- Bar code and label printing software
- Full two year warranty
- **30 Day Money Back Guarantee**
- **Extensive VAR/Dealer Discounts**

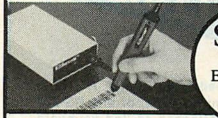
VARIANT MICROSYSTEMS
3140 De La Cruz Blvd.,
Suite 200
Santa Clara, CA 95054
(408) 980-1880

CIRCLE 386 ON READER SERVICE CARD

To Place
Your Ad
Call (201) 503-5115

Bar Coding

EZBarcode™



$289.00 Includes EZBarcode I Printing Software

Bar Code Reader

Model PC-WEDGE Reader $289.00
- No software changes necessary
- For PC/XT/AT or compatible
- Reads Code 39, 2 of 5, UPC, Codabar
- EZBarcode I printing software included

EZBarcode I Printing Program (ordered separately) **$49.00**
- Enter and print Code 39 and UPC-A
- For dot matrix and LaserJet printers
- Vary bar code height, spacing, etc.

EZBarcode II Printing Program ... $99.00
- Keyboard or text file data input
- Prints Code 39, Interleaved 2 of 5, UPC-A
- Automatic incrementing, multiple columns

dBARCODE™ Program prints bar codes from within dBASE III Plus **$99.00**

TimeKeeping Systems, Inc.
12434 D Cedar Road • Cleveland, OH 44106
(216) 229-2579 • Fax: (216) 229-2573

BARCODE BARGAINS

NEW - WANDERWAND™ BARCODE READER

Think of everything you would ever want in a barcode reader and you'll find it in the new ITS 5308 system! Programmable • Handheld, lightweight • Simple connection with standard RS 232 plug • Holds up to 48,127 characters • Discriminates between label codes, time stamp labels • Calculates and dumps scans • Wand included.

ITS 5306 - PC Wand Barcode Reader. Emulates keyboard • Reads codes UPC, Codabar, 3 of 9, 2 of 5 interleaved • Works with IBM PC's and clones • Lowest cost - versatile, high quality!

PCDOS-MSDOS Printing Software. Generate your own labels on your existing Matrix printer • Call for details.

International Technologies & Systems Corp.
635 North Berry Street, Suite C
Brea, CA 92021
714-990-1880

GUARANTEED LOWEST PRICES
Visa and MasterCard Accepted
ITS

CIRCLE 387 ON READER SERVICE CARD

BAR CODE & MAGNETIC STRIPE READERS for the IBM PC & PS/2

Simple & quick installation
No additional software or port
Metal wand & case
Also available:
Bar code printing software
Magnetic stripe encoders
Units for other computers & terminals
GSA pricing available

TPS ELECTRONICS
4047 Transport Street
Palo Alto, CA 94303
Telephone: 415-856-6833
Telex: (Graphnet) 371-9097 TPS PLA
FAX: 415-856-3843

CIRCLE 388 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

BAR CODING

COMPUTER TRAINING

DATA CONVERSION

DATA CONVERSION

PUBLICATIONS

SUPPLIES

TECH MARKETPLACE
HOME OF THE POWER BUYER

PRODUCT CATEGORIES

HARDWARE

ACCESSORY CARDS
COMMUNICATIONS
COMPATIBLES
COMPUTER SYSTEMS
COOLING DEVICES
GENERAL
MASS STORAGE
PERIPHERALS
SECURITY DEVICES
USED EQUIPMENT

SOFTWARE

ARTIFICIAL INTELLIGENCE
BUSINESS
COMMUNICATIONS
DATA BASE MANAGEMENT
DESK TOP PUBLISHING
EDUCATIONAL
ENGINEERING
EXPERT SYSTEMS
FINANCIAL
GENERAL
GRAPHICS
LANGUAGES
MULTI/USER SYSTEMS
NETWORKING
OPERATING SYSTEMS
PROGRAMMERS TOOLS
PUBLIC DOMAIN
SCIENTIFIC
SECURITY DEVICES
STATISTICS
TERMINAL EMULATION
UTILITIES
WORD PROCESSING

MISCELLANEOUS

ACCESSORIES
BAR CODING
BUSINESS OPPORTUNITIES
CAREER OPPORTUNITIES
COMPUTER INSURANCE
COMPUTER TRAINING
DATA CONVERSION
PUBLICATIONS
SOFTWARE DUPLICATION
SUPPLIES

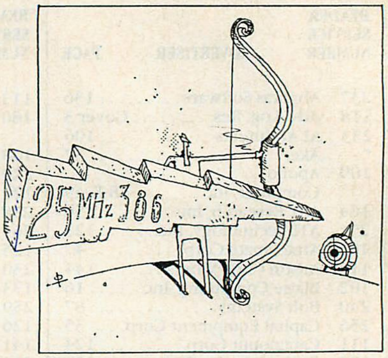
INDEX TO ADVERTISERS

INDEX TO PRODUCTS

JORDENE ZEIMETZ

PROFESSIONAL VIEWPOINT

Special applications and many development efforts demand 25-MHz 386s, while some general applications never will.



While developers and power users are snatching up the latest thunderbolts from the PC heavens (25-MHz 80386-based computers), general users are not, according to an informal *PC Tech Journal* poll.

Nearly half of the respondents say their applications absolutely require these machines; they have special needs that cannot be met at the low end by PCs or at the high end by mainframes. A slightly smaller group—those who are running simple word processing programs and spreadsheets—say they do not need the 25-MHz 386 machines. The remainder have some applications that require them and others that do not.

Not surprisingly, applications that demand this speed are weighted toward computer-aided design, advanced databases and spreadsheets, artificial intelligence, graphics, image processing, and desktop publishing. In all cases, the initial cost takes a back seat to speed requirements.

"It is less expensive to invest in a faster machine than to wait for results," says Robert E. Cain, computer systems scientist at Planning Research Corporation in McLean, Virginia. "Also, the frustration level is lower. Time is money."

While it can be argued that highly computational applications belong on mainframes, the 25-MHz 386 is for anyone who lacks mainframe access or runs computationally intensive graphics applications, for which mainframes are not ideal. "Many of our PCs serve as production centers until large mainframe applications can be developed," says audit manager Bill Yarberry of Enron Corporation in Houston, Texas.

Multiuser environments are also prime targets for the lightning speed of 25-MHz 386 machines—especially as LAN servers. Ross Chevalier, data processing coordinator at the Toronto-based Mercantile United, observes, "In multiuser/network environments, speed is everything. The faster a server can handle I/O requests, open and close files, and build indexes, the better."

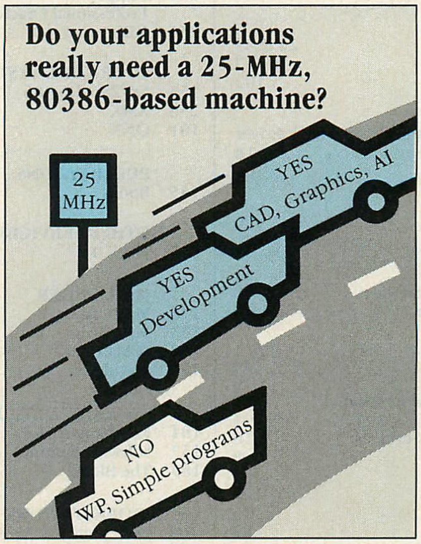
When Structured Query Language (SQL) data-server software (in which the server does most of the processing) becomes available, power will be even more important in this area.

Furthermore, 25-MHz 386s are in demand for software development. Dan Haynes, president of DataTouch Development Corporation in Yukon, Oklahoma, says, "Every second eliminated from compile-and-link time increases productivity and [decreases] programmer burnout."

Senior software engineer Peter Santoro of Gerber Scientific Products Inc. in Manchester, Connecticut, says, "As a software developer and a user of rapid prototyping, speed is very important to me. The more compile-link-edit-test cycles I can go through, the more ideas I can test before committing to a design."

THE NAY-SAYERS

Those who do not need 25-MHz 386 machines fall into three groups: (1) general-application users who get adequate processing power from their PC/XTs, PC/ATs, and compatibles, (2) users and developers who are content with their 10- and 12-MHz 286 machines, and (3) developers and power users who own 16- or 20-MHz 386s.



In this last group, many believe the small performance upgrade to 25-MHz (about 25 percent) is not worth the $3,000 to $5,000 in increased cost because they already have 386 functionality (larger address space and ability to run multiple DOS programs). Many will hold out for 486 machines.

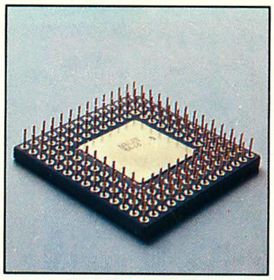
Several respondents point out that their applications are I/O-bound (limited by the speed of the disk), not CPU-bound (limited by the speed of the processor). Therefore, their biggest concerns are faster disks and more memory for disk caching.

"The applications I develop are typically disk-bound," reports Bryan P. Haynes, president of Haynes Software in Costa Mesa, California. "Until disk speeds can increase proportionately, faster machines will mainly benefit CPU-bound applications."

TARGETING THE MARKET

While it has not been long since 25-MHz 386-based PCs started shipping (mid-1988), a large number of developers and users already view them as integral to their specialized needs.

Of those who do not, most developers already own 386 machines, having little incentive to upgrade until the next generation of processors is available. General-application users say their XTs and ATs give them satisfactory performance (at least until OS/2 increases in popularity and a new generation of software emerges). When they eventually replace their PCs, mainstream users are likely to buy lower-end (and lower-cost) 386 machines (including those with the 16-bit 80386SX) and 10- and 12-MHz 286s. For these users, 25 MHz will always be overkill.



Being the first to use the 80386 chip, we established one fact.

The Newspaper for the Microcomputing Community

InfoWorld

Volume 8, Issue 28

July 14, 1986

Special Report: PC, AT Clones
See Page 29.

Fast and Easy Mac Accounting
See Review, Page 49.

Micropro To Market Desktop Program

By Karen Sorensen
and Laurie Flynn
InfoWorld Staff

SAN RAFAEL, CA — Micropro International Corp. announced plans to develop, in conjunction with Island Graphics Corp., a Wordstar-compatible IBM PC desktop publishing program that will operate under Microsoft Corp.'s Windows.

The agreement between Micropro and Island of Sausalito, California, gives Micropro exclusive worldwide retail marketing rights to the product, code-named Prism. Micropro said it expects to ship the product by the end of 1986.

Prism will accept text files from Micropro's Wordstar, Wordstar 2000, and Easy word processing programs, as well as those conforming to IBM's Document Content Architecture protocol, according to H. Glen Haney, president of Micropro. A limited ...itor will be incorporated into the ...ser can manipulate text

Gene Lu, president of Advanced Logic Research (left), and Dave Kirkey, vice president of sales and marketing, display the company's 80386-based prototype.

First Micro To Employ 80386 Chip Due in Fall

By Priscilla M. Chabal
and Elizabeth Ranney
InfoWorld Staff

IRVINE, CA — Hardware maker Advanced Logic Research Inc. said it is poised to beat IBM Corp. and other major players to market with a microcomputer based on Intel Corp.'s next-generation 80386 microprocessor.

The computer, the Access 386, will use a recently announced 80386 BIOS from Phoenix Technologies Ltd. of Norwood, Massachusetts, and will be compatible with the IBM PC AT, according to Gene Lu, Advanced Logic's president. The machine, which is being manufactured for the company in Singapore, will use the MS-DOS 3.2 operating system and allow software written for the current 286 chip to run three to four times faster, the company said. The machine will be equipped with a 40-megabyte hard disk drive and a proprietary video graphics card.

The company said the machine is slated to be on the market by the fourth quarter this year and is expected to be priced between $7,000 and $8,000.

IBM and Compaq Computer Corp., both reported to be working on 80386-specific machines, are not expected to announce them until sometime in 1987.

Although there is no 80386-specific version of MS-DOS that would allow software to take advantage of many of the advanced chips features, Lu said the ...chine will nonetheless outperform an ...peal to users whose programs

Continued on Page 8

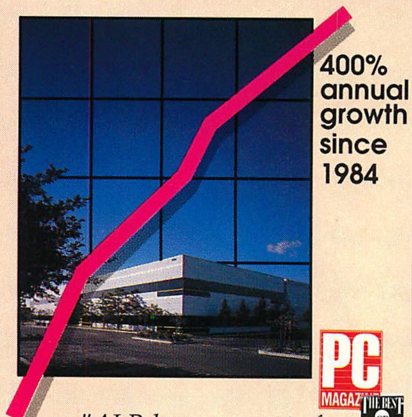
California Legislator Backs Off ...fort to Pass Warranty Bill

... 1987 if she isn't satisfied with the ... efforts by then. ... the bill die, which I see as ... moving in the ...

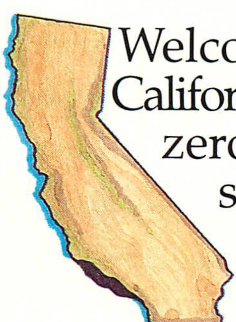
"...you don't have to be based in Texas to build the world's fastest PCs."

PC WEEK

March 29, 1988



"ALR has come out of nowhere over the last two years to earn a spot in the sun among important PC-compatible makers."
- Jim Seymour January 12, 1988



Welcome to California, the zero wait state.

In July of '86 Advanced Logic Research was the world's first manufacturer to release an 80386 based PC. Today, industry peers have recognized both the ALR FlexCache 20386 and 25386 as the world's fastest PCs. These systems are the most recommended tools for your most demanding business applications in network, multi-user and CAD/CAM environments.

NOVELL

TESTED & APPROVED

"The ALR FlexCache 25386 Model 150 earns the highest CPU performance rating ever - 6.46."
Info World - August 15, 1988

INFO WORLD
THE PC NEWS WEEKLY

The ALR FlexCache 20386 Model 150 achieved a rating of 19.7 in Byte Lab. *"Tops in price and performance"*

BYTE LAB *Byte June, 1988*

" The performance they achieve sets a new standard for PCs."

PCResource *June, 1988*

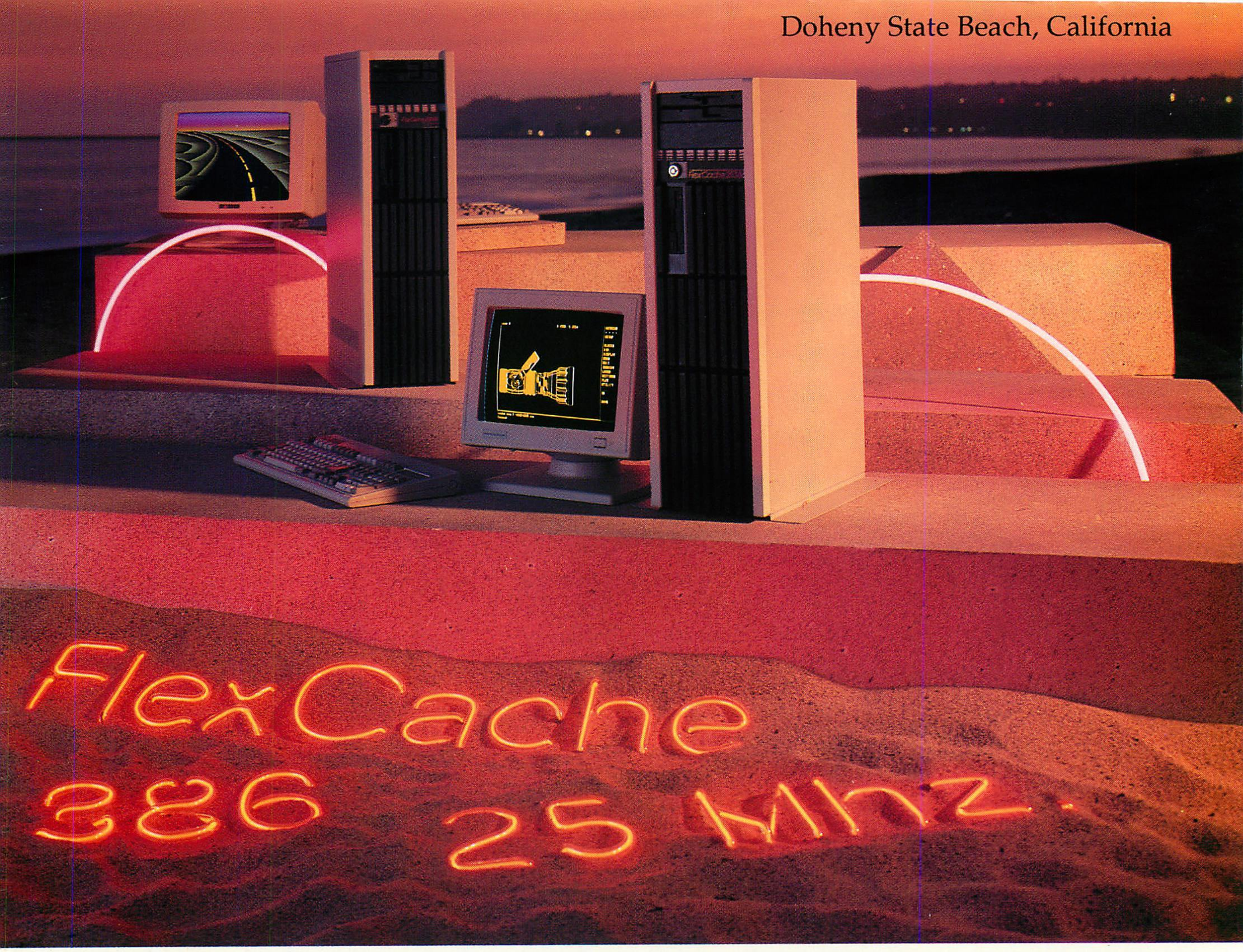
" Well, for once the answer isn't to run right out and get your hands on a Compaq Deskpro 386/20.™ Rather, it's to get your hands on an ALR FlexCache 20386."

First Looks
March 15, 1988
PC MAGAZINE

" ... the FlexCache 20386 yields the highest number of MIPS (million instructions per second) per dollar."

TECH PC JOURNAL *June, 1988*

ALR FlexCache 25386 and 20386 are trademarks of Advanced Logic Research, Inc. Compaq Deskpro 386S is a trademark of Compaq



Redefining Performance and Price

Although we can't go any further west, ALR is claiming new frontiers in 80386 performance and processing speeds. And we're doing it for less. Less than systems that deliver only half the performance using the new Intel 16MHz, 16-bit, 386SX™ microprocessor. Even less than some 80286 based systems. To find out more information on the fastest PCs on the planet and our entire line of 386 products call:

1-800-444-4ALR

for the name and location of the nearest ALR dealer

	ALR FlexCache	Compaq Deskpro	IBM PS/2 50 Z	ALR FlexCache	IBM PS/2 70 386
Models	20386DT-401	386s Mod. 40	8550-061	25386-150	8570-A21
Microprocessor	80386 32-bit 20MHz 32KB, 82385 Memory Cache (35ns)	80386SX 16-bit 16MHz	80286 16-bit 10MHz 0 wait-state	80386 32-bit 25MHz 64KB Extended Emulation 82385 Memory Cache (25ns)	80386 32-bit 25MHz 64KB Memory Cache (30ns)
PC Lab Benchmarks 80286 Instruction Mix - Seconds	**2.75 SEC.**	**4.56 SEC.**	**7.20 SEC.**	**2.20* SEC.** *80386 Instruction Mix	**2.36* SEC.**
Optional Math CoProcessor	80387 20MHz	80387SX16MHz	80287 10MHz	80387 25MHz	80387 25MHz
Memory (RAM)	1MB (80ns)	1MB (100ns)	1MB (85ns)	2MB (60ns)	2MB (80ns)
Storage 1.2MB, 5 1/4" FD 1.44MB, 3 1/2" FD	1 Optional ($225.00)	1 Optional ($245.00)	Not Available 1	1 Optional ($225.00)	Not Available 1
Fixed Disk Std.	40MB (28ms)	40MB (28ms)	60MB (27ms)	150MB (18ms)	120MB (23ms)
Opt. Internal Full Height Fixed Disk	66MB or100MB	No Support	No Support	150MB or 300MB	No Support
Video	Optional 16 bit VGA ($399.00)	16 bit VGA	VGA	16 bit VGA	VGA
Price	$4490.00*	$5199.00*	$4595.00*	$9499.00*	$11,295.00*

* Prices and configurations as of August 1988 and subject to change.

Advanced Logic Research, Inc.
9401 Jeronimo, Irvine, CA 92718
Phone: 714-581-6770 FAX: 714-581-9240
Circle No. 118 for end-user. No. 116 for reseller.

Introducing REMOTE²
It's never been so easy to do so much in distant PC operation.

Until now, to have this kind of flexibility and control over a host PC, you'd have to be in the same room.

Now, even if you're thousands of miles away, REMOTE² allows you to operate a host PC's application software with total control and exact mapping of the host keyboard...fast file transfers even while an application program is running ...remote printer redirection...an error-checked, data-compressed link even with conventional modems...and CGA color graphics.

REMOTE² comes in two parts–R2HOST and R2CALL–available together or separately, so you can create the combination to meet your exact needs. R2HOST is also accessible from most terminals and terminal emulators.

REMOTE² is packed with features users have asked for. A choice of three distinct automatic and manual answering modes. Directory-to-directory file transfers using a half-screen display of host files. Proprietary file transfer protocol with redundant file skipping and partial file recovery (other popular protocols also supported). A "Phone Book" that facilitates one-entry calls from listing of names, numbers, and passwords. Host call-back capability. Integrated, context-sensitive help system. LAN access. Mainframe access to an IBM host with IRMA. And more.

Discover the new remote control program from the makers of CROSSTALK. Ask your dealer about REMOTE² or write us.

REMOTE²™

CROSSTALK COMMUNICATIONS/1000 Holcomb Woods Parkway, Roswell, Georgia 30076/(800) 241-6393

A Division of Digital Communications Associates, Inc.

CIRCLE NO. 159 ON READER SERVICE CARD